Aplikacija za računanje determinante u programskom jeziku C

Čeč, Lorena

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:602787

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni preddiplomski studij

APLIKACIJA ZA RAČUNANJE DETERMINANTE U PROGRAMSKOM JEZIKU C

Završni rad

Lorena Čeč

Osijek, 2023.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. PREGLED POSTOJEĆIH RJEŠENJA 2

2.1. Web stranica *Symbolab* 2

2.2. Mobilna aplikacija *Matrix Determinant Calculator* 3

3. DETERMINANTA 5

3.1 Svojstva determinanti 5

3.2. Izračun determinante 6

3.2.1. Sarrusovo pravilo 7

3.2.2. Laplaceov razvoj 9

3.2.3. LU dekompozicija 11

3.2.4. Gaussova eliminacija 12

3.3. Primjena determinante 13

4. ALATI ZA IZRADU APLIKACIJE 15

4.1. Programski jezik C# 15

4.2. Visual Studio 15

4.3. Windows Forms 16

5. IZRADA APLIKACIJE 20

5.1. Obrazac *LaunchScreen* 21

5.2. Obrazac *LearnMore* 23

5.3. Obrasci *Learn* 25

5.4. Obrazac *Calculation* 26

5.4.1. Funkcije za računanje determinante 31

6. PRIKAZ KORIŠTENJA APLIKACIJE 35

7. ZAKLJUČAK 42

LITERATURA 43

SAŽETAK.....45
ABSTRACT.....46

1. UVOD

Determinanta matrice je broj specifično definiran samo za kvadratne matrice. Ovaj broj je vrlo koristan u linearnoj algebri te stoga je bitno znati kako ga odrediti i gdje ga sve možemo koristiti. Cilj ovoga rada je pobliže objasniti važnost determinante te omogućiti računanje determinante za danu matricu putem jednostavne aplikacije.

Na početku rada napravljen je pregled već postojećih programskih rješenja. Prikazana su dva primjera odnosno web stranica i mobilna aplikacija koje su uspoređene sa samim zadatkom ovog završnog rada. Potom je opisano što je determinanta, njena svojstva i gdje se koristi. Također, objašnjeni su mogući načini rješavanja determinante i razlike među njima. Za lakše razumijevanje određivanja determinante riješeno je i nekoliko primjera kako bi se dodatno istaknule razlike u rješavanju determinanti. Samim primjerima istaknuta je i kompliciranost određivanja determinanti matrica viših redova što daje dodatan razlog za stvaranje aplikacije za rješavanje determinante.

Opisani su svi alati korišteni za izradu aplikacije. Tu su uključeni programski jezik C#, Visual Studio u kojem je napisan programski kod te predložak unutar Visual Studija tj. Windows Forms pomoću kojega je napravljena aplikacija za računanje determinante. U opisu su spomenute osnovne značajke pojedine tehnologije, a uz to dodane su i neke od zanimljivosti te opće informacije o navedenim tehnologijama.

Prikazan je postupak rješavanja zadatka završnog rada te je detaljno opisan svaki korak pri stvaranju funkcionalnog koda i aplikacije. Svaki je obrazac objašnjen kroz prikaz svih funkcija obrasca, pojašnjenje tih funkcija, opis izgleda obrasca i njegove funkcionalnosti. Posljednje, prikazan je konačan izgled aplikacije i riješeni primjeri unutar aplikacije za matrice četvrtog, osmog i desetog reda.

1.1. Zadatak završnog rada

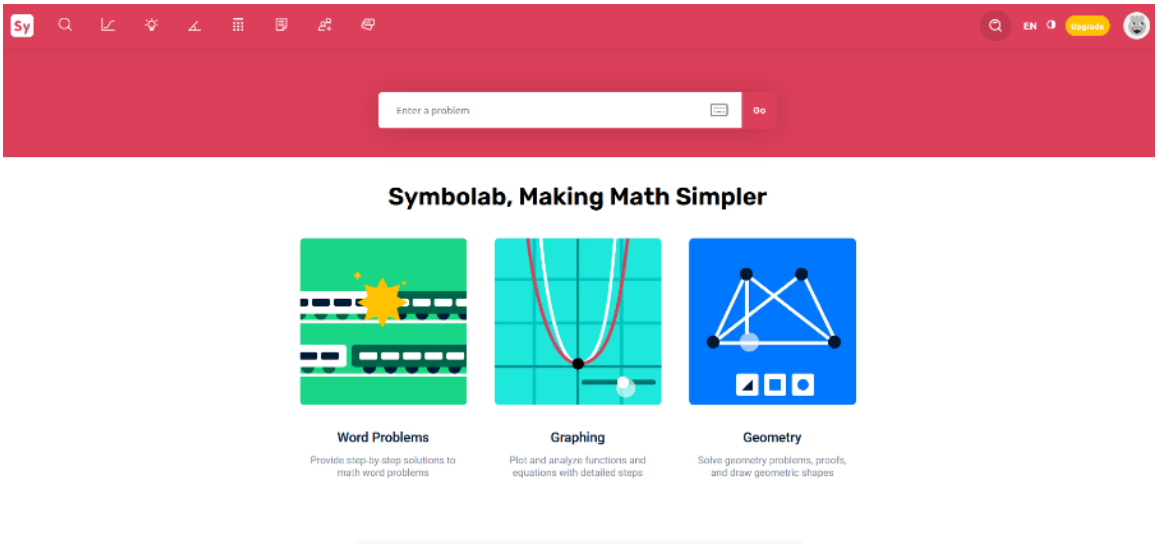
Zadatak završnog rada je izraditi aplikaciju u programskom jeziku C# koja će za zadanu matricu izračunati njenu determinantu. Aplikacija mora omogućiti korisniku da unese red matrice i potom unosi sve elemente matrice toga reda. Nakon unosa, aplikacija treba korisniku prikazati iznos determinante i ponuditi mogućnost upisivanja nove matrice.

2. PREGLED POSTOJEĆIH RJEŠENJA

Prilikom pretrage aplikacija i web stranica za rješavanje determinante ponuđeno je mnogo programskih rješenja sličnih aplikaciji koja je izrađena u ovom završnom radu. Postoje kompleksne stranice koje pružaju i više od samih rezultata pošto korisniku prikazuju i postupak rješavanja determinante korak po korak, a isto tako omogućuju i rješavanje drugih matematičkih problema. Također postoje i jednostavnija rješenja sa samo jednom namjenom odnosno isključivo rješavaju determinantu.

2.1. Web stranica *Symbolab*

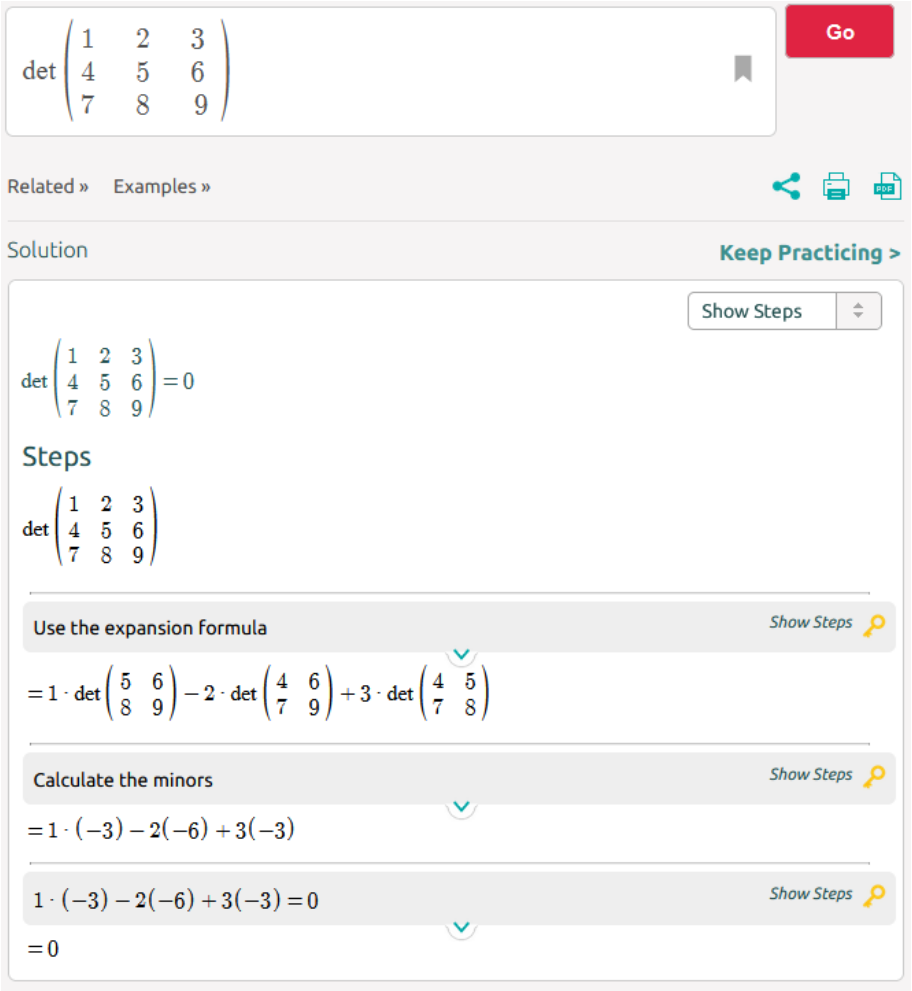
Web stranica Symbolab pruža korisnicima razne funkcionalnosti u mnogim područjima. Korisnici ove stranice mogu crtati grafove, rješavati jednadžbe uz prikaz rješenja korak po korak te čak i samostalno vježbati na raznim primjerima i kvizovima. Početni izgled web stranice prikazan je na slici 2.1.



Slika 2.1. Početni izgled web stranice *Symbolab*

Prema slici 2.1. ova web stranica se možda čini jednostavna, ali klikom na bilo koju opciju u navigacijskoj traci korisniku se nudi širok izbor primjera i kalkulatora i to ne samo u području matematike, nego i fizike, kemije, a i nekih svakodnevnih područja poput financija, fitnessa pa čak i kulinarstva. Pošto Symbolab pruža mnoge mogućnosti korisnicima, mnoge od tih mogućnosti nisu u potpunosti besplatne. Iako ova web stranica ima mogućnost rješavanja determinante matrice

korak po korak, ta opcija je besplatna samo za matrice do 3. reda dok za veće matrice prikazuje samo rješenje. [1] Prikaz rješenja determinante matrice 3. reda na ovoj web stranici prikazan je na slici 2.2.



Slika 2.2. Rješavanje determinante na web stranici *Symbolab*

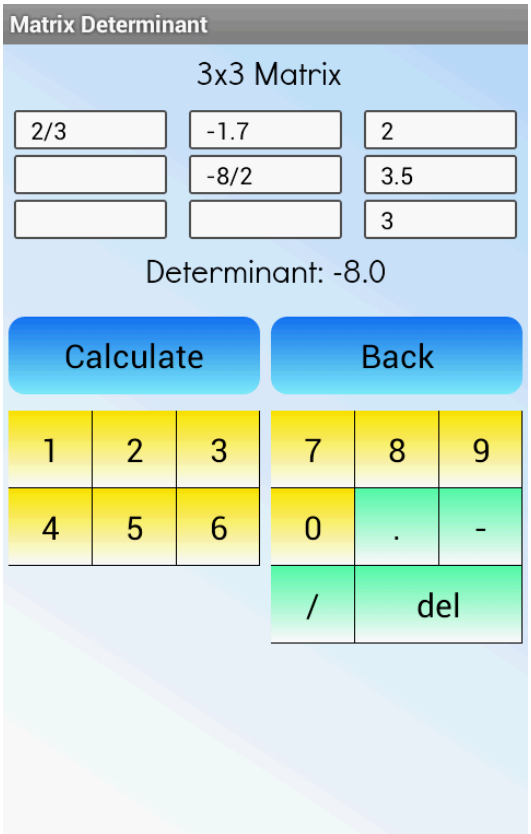
Symbolab postoji i u obliku mobilne aplikacije koja omogućuje slične funkcionalnosti kao web stranica, no dodatno pruža korisniku opciju rješavanja problema kroz fotografiranje zadataka. Također, ni mobilna aplikacija nije u potpunosti besplatna. [2]

2.2. Mobilna aplikacija *Matrix Determinant Calculator*

Većina web stranica za rješavanje determinante ima i dodatne funkcionalnosti poput mogućnosti zbrajanja matrica, računanja inverza matrice i sličnih operacija, dok su mobilne i desktop aplikacije često napravljene isključivo za rješavanje determinanti.

Jedna od mobilnih aplikacija za računanje determinante je *Matrix Determinant Calculator* čiji je početni izgled prikazan na slici 2.3. dok je na slici 2.4. prikazan primjer rješavanja determinante.

Slika 2.3. Početni izgled aplikacije *Matrix Determinant Calculator*



Slika 2.4. Primjer rješavanja determinante u aplikaciji *Matrix Determinant Calculator*

Kao što je prikazano na slici 2.3. ova aplikacija, kao i mnoge druge aplikacije, isključivo je stvorena samo za rješavanje determinante bez ikakvih dodatnih informacija ili uputa za rješavanje. [3] Stoga je u aplikaciji koja je izrađena u ovom završnom radu dodan informativni dio koji je napravljen za korisnike koji žele znati više o determinantama i koji žele sami naučiti rješavati determinante.

3. DETERMINANTA

Determinanta se može definirati kao skalarna veličina dobivena zbrojem umnožaka elemenata kvadratne matrice i njihovih kofaktora prema propisanom pravilu.

Općenito, determinanta matrice *A* označava se kao $det(A)$. Važno je napomenuti da se elementi matrice upisuju unutar uglatih zagrada dok se za izračun determinante elementi zapisuju unutar ravnih zagrada.

Npr. ako je zadana matrica 2. reda njena determinanta označava se kao: $det(A) = |A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$

3.1 Svojstva determinanti

Za bolje razumijevanje determinante bitno je znati njena svojstva koja znatno mogu pomoći pri rješavanju determinante. Ovo se pogotovo odnosi na svojstva gdje se automatski može prepoznati da je determinanta jednaka nuli (svojstva broj 1. i 6.).

Neka od svojstava determinanti navedena su u nastavku.

1. Determinanta matrice koja sadrži dva jednaka stupca jednaka je nuli.

2. Determinanta će promijeniti predznak ako se zamjene dva stupca.

3. Ukoliko se radi s trokutastim matricama, determinanta matrice bit će jednaka umnošku elemenata na dijagonali. Za matricu *A* vrijedit će:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix} \rightarrow |A| = a_{11}a_{22} \dots a_{nn} \tag{3-1}$$

Na primjer, jedinična matrica na dijagonali sadrži samo jedinice stoga vrijedi da je determinanta jedinične matrice jednaka 1 odnosno $\det(I) = 1$.

4. Determinanta se neće promijeniti ako se jednom stupcu matrice doda linearna kombinacija ostalih stupaca.

5. Za matrice *A* i *B* vrijedit će izraz:

$$\det(AB) = \det(A) \cdot \det(B). \tag{3-2}$$

Za matricu *A* pod uvjetom da je regularna vrijedit će izraz:

$$\det(AA^{-1}) = \det(A) \cdot \det(A^{-1}) = \det(I) = 1 \tag{3-3}$$

Prethodni izraz povlači za sobom da je determinanta inverzne matrice A^{-1} obrnuto proporcionalna determinanti matrice *A*.

Odnosno, vrijedi:

$$\det(A^{-1}) = \frac{1}{\det(A)} \tag{3-4}$$

6. Determinanta će biti različita od nule ako i samo ako je zadana matrica regularna odnosno stupci matrice moraju biti linearno nezavisni.

7. Što se tiče odnosa determinante matrice *A* i determinante transponirane matrice *A* tj. matrice A^T, vrijedi izraz:

$$\det(A) = \det(A^T) \tag{3-5}$$

Sva navedena svojstva vrijede i za stupce i za retke matrice. [4]

3.2. Izračun determinante

Determinanta matrice *n*-tog reda može se definirati uz pomoć determinante matrice čiji je red jednak *n* - 1.

Za determinantu matrice prvog reda vrijedi:

$$A = [a] \rightarrow \det(A) = a \tag{3-6}$$

Za matricu drugog reda $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ vrijedi da je njena determinanta:

$$\det(A) = a_{11}a_{22} - a_{12}a_{21} \tag{3-7}$$

Za matricu trećeg reda $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ vrijedi da je njena determinanta:

$$\det(A) = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{21}\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} + a_{31}\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \tag{3-8}$$ [5]

Primjer 3.1. Rješavanje determinante matrica prvog, drugog i trećeg reda.

a) $A = [3] \rightarrow \det(A) = 3$

b) $A = \begin{bmatrix} 3 & -12 \\ 8 & -5 \end{bmatrix} \rightarrow \det(A) = -15 + 96 = 81$

c) $A = \begin{bmatrix} 5 & 4 & 3 \\ 3 & 4 & 1 \\ 1 & 2 & 2 \end{bmatrix} \rightarrow \det(A) = 5\begin{vmatrix} 4 & 1 \\ 2 & 2 \end{vmatrix} - 3\begin{vmatrix} 4 & 3 \\ 2 & 2 \end{vmatrix} + 1\begin{vmatrix} 4 & 3 \\ 4 & 1 \end{vmatrix} = 16$

Determinanta matrice *n*-tog reda $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$ jednaka je sljedećem broju:

$$\begin{aligned} \det(A) &= a_{11}\det(A_{11}) - a_{21}\det(A_{21}) + \dots + (-1)^{n+1} a_{n1}\det(A_{n1}) \\ &= \sum_{k=1}^{n} (-1)^{k+1} a_{k1}\det(A_{k1}) \end{aligned} \tag{3-9}$$

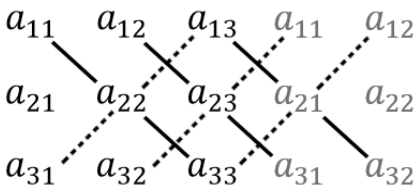
Računanje determinante pomoću definicije je korisno kada se računa sa matricama drugog ili trećeg reda, ali kada se radi o matricama višeg reda takav način računanja je vrlo kompliciran i lako se mogu dogoditi greške pri računanju. Stoga postoje razni načini rješavanja determinanti pomoću kojih se determinanta brže izračuna, a neke od njih su Sarrusovo pravilo, Laplaceov razvoj, LU dekompozicija itd.

3.2.1. Sarrusovo pravilo

Sarrusovo pravilo se koristi za izračun determinante matrice trećeg reda. Ovo pravilo nazvano je po francuskom matematičaru Pierreu Frédéricu Sarrusu (1798. – 1861.). [6]

Ako je zadana matrica $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$, njezina se determinanta može izračunati prema shemi prikazanoj na slici 3.1.

Prva dva stupca matrice prepišu se desno od trećeg stupca, dajući pet stupaca u redu. Zatim se zbroje umnošci dijagonala koje idu odozgo prema dolje (pune linije na slici 3.1.) i oduzmu se umnošci dijagonala koje idu odozdo prema gore (isprekidane linije na slici 3.1.).



Slika 3.1. Shema za računanje determinante pomoću Sarrusovog pravila

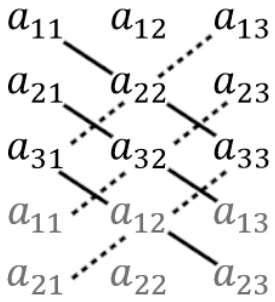
Dakle matricu *A* se može zapisati kao:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{matrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{matrix}$$

Prateći sliku 3.1. dobije se da je determinanta matrice *A* jednaka:

$$\det(A) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12} \tag{3-10}$$

Analogno pravilo dobit će se ako se ispod determinante prepišu njezina prva dva retka. [7] Takva je shema prikazana na slici 3.2.



Slika 3.2. Shema za računanje determinante pomoću Sarrusovog pravila

Primjer 3.2. Računanje determinante pomoću Sarrusovog pravila.

$$A = \begin{bmatrix} 1 & 6 & 3 \\ 3 & 1 & 2 \\ 2 & 4 & 3 \end{bmatrix} \rightarrow A = \begin{bmatrix} 1 & 6 & 3 \\ 3 & 1 & 2 \\ 2 & 4 & 3 \end{bmatrix} \begin{matrix} 1 & 6 \\ 3 & 1 \\ 2 & 4 \end{matrix}$$

$$\det(A) = 1 \cdot 1 \cdot 3 + 6 \cdot 2 \cdot 2 + 3 \cdot 3 \cdot 4 - 2 \cdot 1 \cdot 3 - 4 \cdot 2 \cdot 1 - 3 \cdot 3 \cdot 6$$

$$\det(A) = (3 + 24 + 36) - (6 + 8 + 54)$$

$$\det(A) = -5$$

3.2.2. Laplaceov razvoj

Laplaceov razvoj može se provesti po bilo kojem retku ili stupcu matrice. Ako je zadana matrica *A* *n*-tog reda te ako se u toj matrici izostavi *i*-ti redak i *j*-ti stupac, dobit će se matrica čija se determinanta naziva subdeterminanta ili minora. U primjeru 3.3. može se primijetiti da se tijekom rješavanja determinante zapravo stvara velik broj subdeterminanti koje na kraju pomažu u pronalasku samog rezultata rješavanja.

Za matricu *A* postoji razvoj:

- po *i*-tom stupcu:

$$\det(A) = \sum_{k=1}^{n} (-1)^{k+i} a_{ki} \det(A_{ki}), \tag{3-11}$$

- po *i*-tom retku:

$$\det(A) = \sum_{k=1}^{n} (-1)^{k+i} a_{ik} \det(A_{ik}). \tag{3-12}$$ [8]

Primjer 3.3. Izračun determinante matrice *A* pomoću Laplaceovog razvoja.

$$A = \begin{bmatrix} 3 & 2 & 5 & 1 \\ 4 & 1 & 3 & 7 \\ 1 & 2 & 4 & 1 \\ 3 & 6 & 4 & 2 \end{bmatrix}$$

Za zadanu matricu odabran je razvoj po 3. redu pošto se u njemu nalazi najviše jedinica. Isto tako mogao se izabrati i razvoj po 4. stupcu zbog istog broja jedinica. Općenito, korisno je znati da se pri rješavanju determinante najčešće uzimaju stupci tj. reci s nulama jer umnožak determinante s nulom uvijek će biti nula čime se smanjuje dio računanja koji se treba odraditi. Ukoliko matrica ne sadrži niti jednu nulu, uzimaju se stupci ili reci s najvećim brojem jedinica što zapravo uključuje i 1 i -1 jer su to također brojevi s kojima je jednostavno računati. Naravno, pri računanju determinante može se izabrati bilo koji stupac tj. redak, ali bitno je znati da postoje lakši i teži stupci za rješavanje čime se može smanjiti vrijeme rješavanja.

$$\det(A) = (-1)^{3+1} \cdot 1 \cdot \begin{vmatrix} 2 & 5 & 1 \\ 1 & 3 & 7 \\ 6 & 4 & 2 \end{vmatrix} + (-1)^{3+2} \cdot 2 \cdot \begin{vmatrix} 3 & 5 & 1 \\ 4 & 3 & 7 \\ 3 & 4 & 2 \end{vmatrix} + (-1)^{3+3} \cdot 4 \cdot \begin{vmatrix} 3 & 2 & 1 \\ 4 & 1 & 7 \\ 3 & 6 & 2 \end{vmatrix}$$
$$+ (-1)^{3+4} \cdot 1 \cdot \begin{vmatrix} 3 & 2 & 5 \\ 4 & 1 & 3 \\ 3 & 6 & 4 \end{vmatrix}$$

$$det(A) = 1 \cdot \left(2 \begin{vmatrix} 3 & 7 \\ 4 & 2 \end{vmatrix} - 1 \begin{vmatrix} 5 & 1 \\ 4 & 2 \end{vmatrix} + 6 \begin{vmatrix} 5 & 1 \\ 3 & 7 \end{vmatrix}\right) - 2 \cdot \left(3 \begin{vmatrix} 3 & 7 \\ 4 & 2 \end{vmatrix} - 4 \begin{vmatrix} 5 & 1 \\ 4 & 2 \end{vmatrix} + 3 \begin{vmatrix} 5 & 1 \\ 3 & 7 \end{vmatrix}\right) + 4 \cdot \left(3 \begin{vmatrix} 1 & 7 \\ 6 & 2 \end{vmatrix} - 4 \begin{vmatrix} 2 & 1 \\ 6 & 2 \end{vmatrix} + 3 \begin{vmatrix} 2 & 1 \\ 1 & 7 \end{vmatrix}\right) - 1 \cdot (3 \begin{vmatrix} 1 & 3 \\ 6 & 4 \end{vmatrix} - 4 \begin{vmatrix} 2 & 5 \\ 6 & 4 \end{vmatrix} + 3 \begin{vmatrix} 2 & 5 \\ 1 & 3 \end{vmatrix})$$

$$det(A) = 1 \cdot (-44 - 6 + 192) - 2 \cdot (-66 - 24 + 96) + 4 \cdot (-120 + 8 + 39) - 1 \cdot (-42 + 88 + 3)$$

$$det(A) = 142 + -12 + -292 + -49$$

$$det(A) = -211$$

Upravo ovakav pristup koristit će se u programskom kodu ovog završnog rada koji će detaljno biti objašnjen u 5. poglavlju rada.

3.2.3. LU dekompozicija

U linearnoj algebri, LU (engl. *lower/upper*) dekompozicija faktorizira matricu kao umnožak donje trokutaste matrice i gornje trokutaste matrice. U donje trokutastoj matrici svi elementi iznad dijagonale su nula, a u gornje trokutastoj matrici svi elementi ispod dijagonale su nula. [9] Na primjer, za matricu trećeg reda, njezina LU dekompozicija izgleda ovako:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \tag{3-13}$$

Primjer 3.4. Računanje LU dekompozicije za matricu *A* pomoću sustava linearnih jednadžbi i rješavanje determinante matrice *A*.

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = \mathrm{LU} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Ovaj problem može se svesti na rješavanje sustava linearnih jednadžbi.

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = \begin{bmatrix} \mathrm{u}_{11} & \mathrm{u}_{12} & \mathrm{u}_{13} \\ \mathrm{l}_{21}\mathrm{u}_{11} & \mathrm{l}_{21}\mathrm{u}_{12} + \mathrm{u}_{22} & \mathrm{l}_{21}\mathrm{u}_{13} + \mathrm{u}_{23} \\ \mathrm{l}_{31}\mathrm{u}_{11} & \mathrm{l}_{31}\mathrm{u}_{12} + \mathrm{l}_{32}\mathrm{u}_{22} & \mathrm{l}_{31}\mathrm{u}_{13} + \mathrm{l}_{32}\mathrm{u}_{23} + \mathrm{u}_{33} \end{bmatrix}$$

Prvo se ispiše ono što se odmah može promijeniti bez rješavanja sustava linearnih jednadžbi.

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 6 + \mathrm{u}_{22} & 12 + \mathrm{u}_{23} \\ 2 & 4 + \mathrm{l}_{32}\mathrm{u}_{22} & 8 + \mathrm{l}_{32}\mathrm{u}_{23} + \mathrm{u}_{33} \end{bmatrix}$$

Zatim se riješe dobivene linearne jednadžbe.

$6 + \mathrm{u}_{22} = 8 \rightarrow \mathrm{u}_{22} = 2$

$12 + \mathrm{u}_{23} = 14 \rightarrow \mathrm{u}_{23} = 2$

$4 + \mathrm{l}_{32}\mathrm{u}_{22} = 6 \rightarrow 2\,\mathrm{l}_{32} = 2 \rightarrow \mathrm{l}_{32} = 1$

$8 + \mathrm{l}_{32}\mathrm{u}_{23} + \mathrm{u}_{33} = 13 \rightarrow \mathrm{u}_{33} = 13 - 8 - 2 = 3$

Ubacivanjem rješenja linearnih jednadžbi dobiju se sljedeće matrice, a samim time i rješenje:

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix}$$

Pošto vrijedi da je $A = LU$, onda je $det(A) = det(L)\,det(U)$. Budući da su L i U trokutaste matrice, pronalaženje njihove determinante je vrlo jednostavno. Kao što je napisano u poglavlju 3.1. u svojstvima determinante - ako se radi s trokutastim matricama, determinanta matrice bit će jednaka umnošku elemenata na dijagonali. Stoga determinanta matrice *A* bit će jednaka:

$$det(A) = (1 \cdot 1 \cdot 1) \cdot (1 \cdot 2 \cdot 3) = 6$$

3.2.4. Gaussova eliminacija

Neka je zadana matrica:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Kao što je spomenuto u svojstvima determinanti, jednom retku matrice može se dodati drugi redak koji je pomnožen s brojem različitim od nule. Primjenjujući ovo svojstvo, matricu *A* se može svesti na gornje trokutasti oblik.

Npr. drugi redak pomnožen s 2 doda se prvom retku:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} / \cdot 2$$

$$A = \begin{bmatrix} 2a_{11}a_{21} & 2a_{12}a_{22} & 2a_{13}a_{23} \\ 0 & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Ovaj postupak naziva se Gaussova eliminacija. [10]

Primjer 3.5. Računanje donje i gornje trokutaste matrice za matricu *A* pomoću Gaussove eliminacije i rješavanje determinante matrice *A*.

$$A = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 0 \\ 4 & 4 & 1 \end{bmatrix}$$

Prvo se oduzme redak 1 pomnožen s 2 od retka 2: $\mathrm{R2} = \mathrm{R2} - 2\mathrm{R1}$

$$U = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & -4 \\ 4 & 4 & 1 \end{bmatrix}$$

Kako bi dobili sve nule ispod dijagonale, oduzme se red 1 pomnožen s 4 od retka 3: $\mathrm{R3} = \mathrm{R3} - 4\mathrm{R1}$

$$U = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & -4 \\ 0 & 0 & -7 \end{bmatrix}$$

Odgovarajuća donje trokutasta matrica L imat će jedinice duž glavne dijagonale, a elementi ispod glavne dijagonale dobiveni su iz odgovarajućih inverznih operacija.

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix}$$

Determinanta matrice *A* bit će jednaka:

$$det(A) = (1 \cdot 1 \cdot 1) \cdot \big(1 \cdot 1 \cdot (-7)\big) = -7$$

3.3. Primjena determinante

Pomoću determinante može se pronaći inverz matrice, a koristi se i za rješavanje linearnih jednadžbi putem inverzije matrica. Izraz za inverz matrice glasi:

$$A^{-1} = \frac{1}{\det(A)} adj(A) \qquad \text{(3-14)[11]}$$

(3-14) [11]

Također, preko izraza za izračun determinante lako se može zapamtiti unakrsni umnožak dvaju vektora odnosno:

$$a \times b = \begin{vmatrix} i & j & k \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix} i - \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix} j + \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} k \qquad \text{(3-15)}$$

Preko determinante može se dobiti i mnogo informacija o danoj matrici, posebice o njezinu rangu. Rang matrice A, RK(A), je broj redaka ili stupaca, n, najveće $n \times n$ kvadratne podmatrice od A za koju je determinanta različita od nule. [12]

Primjer 3.6. Računanje ranga matrice *A*

a) $A = \begin{bmatrix} 3 & -12 \\ 8 & -5 \end{bmatrix} \rightarrow \det(A) = -15 + 96 = 81 \rightarrow \det(A) \neq 0$

Pošto je determinanta različita od nule rang matrice A je $RK(A) = 2$

b) $A = \begin{bmatrix} 2 & 6 \\ 4 & 12 \end{bmatrix} \rightarrow \det(A) = 24 - 24 = 0$

Pošto je determinanta jednaka nuli rang matrice A je $RK(A) = 1$

Što se tiče područja van linearne algebre, u fizici primjena matrica i determinanti pomaže u proučavanju električnih krugova, otporničkoj pretvorbi električne energije u drugu energiju i optici. Štoviše, pomaže izračunati izlaznu snagu baterije, kvantnu mehaniku itd. Stoga matrice i determinante igraju bitnu ulogu u izračunima, posebno kod Kirchhoffovih zakona. U matematici se koristi za računanje površine trokuta, površine paralelograma i volumena paralelopipeda. [13]

4. ALATI ZA IZRADU APLIKACIJE

Pri izradi aplikacije za računanje determinante korišten je programski jezik C#, program Visual Studio 2019 te Windows Forms odnosno jedan od predložaka dostupan unutar Visual Studija u kojem je izrađena sama aplikacija za računanje determinante.

4.1. Programski jezik C#

C# je moderan, objektno orijentiran programski jezik. On omogućuje izradu raznovrsnih aplikacija koje se izvode u .NET-u. Pošto C# ima svoje korijene u C obitelji jezika, poznat je C, C++, Java i JavaScript programerima. Što se tiče upotrebe, C# se koristi za razvoj web aplikacija, desktop aplikacija, mobilnih aplikacija, igara i još mnogo toga.

C# je 2000. godine razvio Anders Hejlsberg, danski softverski inženjer koji je sudjelovao u stvaranju pregršt pouzdanih programskih alata i jezika, uključujući Microsoftov TypeScript i Delphi.

U studenom 2022. C# je bio na 4. mjestu na PYPL indeksu popularnosti programskih jezika, odmah iza Jave i JavaScripta. Podaci korišteni za sastavljanje ovog indeksa temelje se na tome koliko često ljudi pretražuju upute o različitim programskim jezicima na Googleu. C# se također dosljedno pojavljuje u prvih deset programskih jezika u TIOBE indeksu, izvješću koje svoje podatke izvlači iz kompilacije popularnih tražilica uključujući Google, YouTube i Bing.

U početku se C# izvorno trebao zvati COOL, akronim koji je označavao *C-like Object Oriented Language*. Nažalost, Microsoft nije mogao zadržati ovo ime iz razloga povezanih sa zakonom o zaštitnim znakovima. C# je prvotno dizajniran kao konkurencija Javi, a uzimajući u obzir brzi porast popularnosti i pozitivan odgovor novih i iskusnih programera, može se reći da je taj cilj postignut. [14]

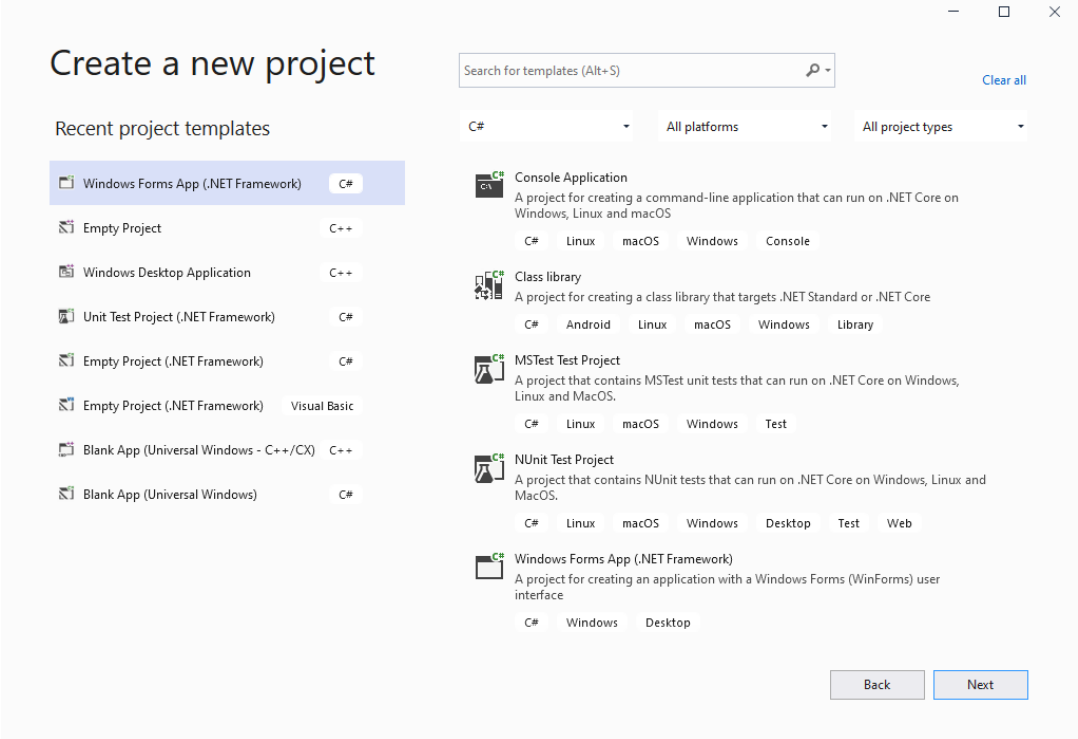
4.2. Visual Studio

Visual Studio je program koji se može koristiti za uređivanje koda, uklanjanje pogrešaka te izradu koda, a potom i objavu same aplikacije. Osim standardnog uređivača i programa za ispravljanje pogrešaka koje nudi većina IDE-a, Visual Studio uključuje prevoditelje, alate za dovršavanje koda,

grafičke dizajnere i mnoge druge značajke za poboljšanje procesa razvoja softvera. Visual Studio je najbrži IDE za produktivnost. Može raditi na bilo kojoj platformi te najčešće se koristi u izradi web stranica, mobilnih aplikacija te web aplikacija i usluga. [15]

Pri pisanju programskog koda, Visual studio dodatno olakšava izradu funkcionalnog koda tako što korisniku nudi različite opcije koda koji može koristiti. Ponuđeni kod može poboljšati rad aplikacije, može pomoći korisniku pri uklanjanju pogrešaka, a također može dati ideju korisniku za pristup drugačiji od zamišljenoga.

Visual Studio pri pokretanju daje korisniku širok izbor mogućnosti predložaka za različite programske jezike koje podržava. Neki od spomenutih predložaka specifično za C# prikazani su na slici 4.1. Na slici je također i predložak koji će se koristiti u ovom završnom radu – Windows Forms.

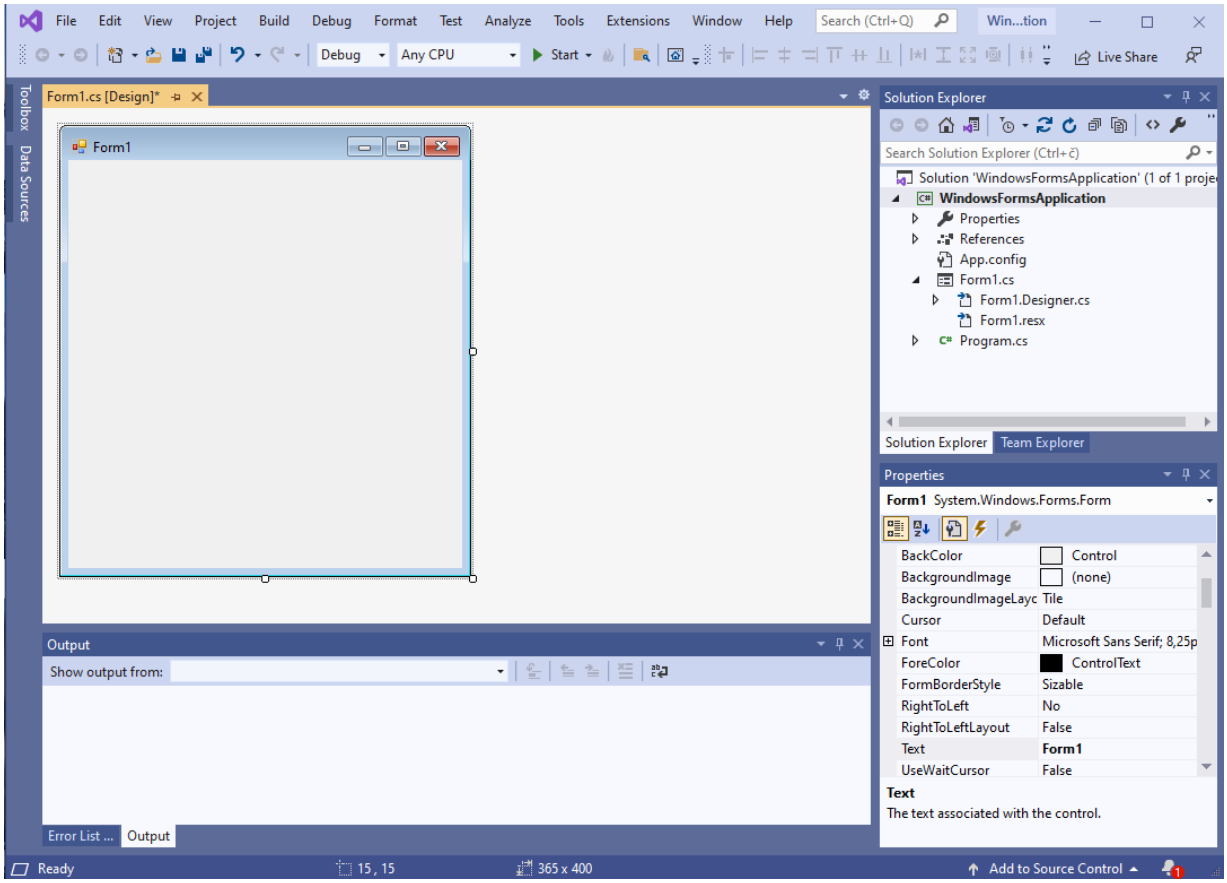


Slika 4.1. Prikaz predložaka za programski jezik C#

4.3. Windows Forms

Windows Forms skup je upravljanih biblioteka u .NET Frameworku dizajniranih za izradu Windows desktop aplikacija. Jedan je od najproduktivnijih načina za stvaranje desktop aplikacija na temelju vizualnog dizajnera koji nudi Visual Studio. [16] Funkcionalnost kao što je postavljanje kontrola povlačenjem i ispuštanjem (engl. *drag and drop*) olakšava stvaranje aplikacija. Kada se

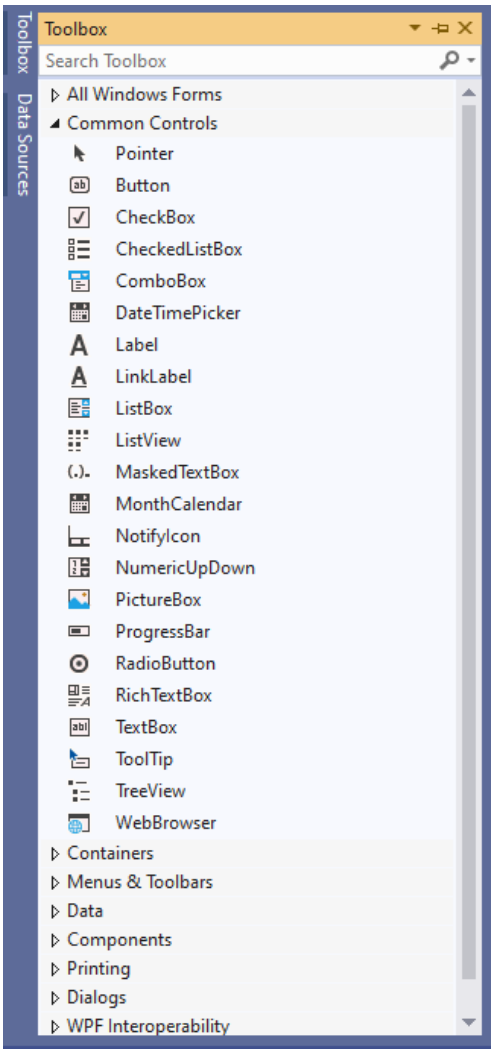
pokrene Visual Studio i odabere Windows Forms, projekt će izgledati kao što je prikazano na slici 4.2.



Slika 4.2. Prikaz novog projekta tj. praznog obrasca

Ukoliko se obrati pozornost na sliku 4.2. može se uočiti da se s lijeve strane nalazi sekcija s alatima (engl. t*oolbox*) koja sadrži sve moguće elemente obrasca poput gumbova (engl. *button*), naziva (engl. l*abel*), prozora za upisivanje teksta (engl. *textbox*), prozora za slike (engl. *imagebox*), itd. Dio takvih alata prikazan je na slici 4.3. gdje su prikazani najkorišteniji alati u obrascima.

Na desnoj strani slike 4.2. može se primijetiti da postoje i postavke za uređivanje samog obrasca (engl. *properties*), a kako se dodaju različiti elementi u obrazac tako na istom mjestu možemo i njihove postavke mijenjati. Među postavkama može se mijenjati boja elementa, font, poravnanje, ime, veličina i mnoge druge stavke. Također, klikom na ikonu munje prikazani su svi događaji (engl. *events*) odnosno ispisane su funkcije koje se mogu primijeniti na obrazac ili element obrasca.



Slika 4.3. Prikaz najčešćih elemenata obrasca

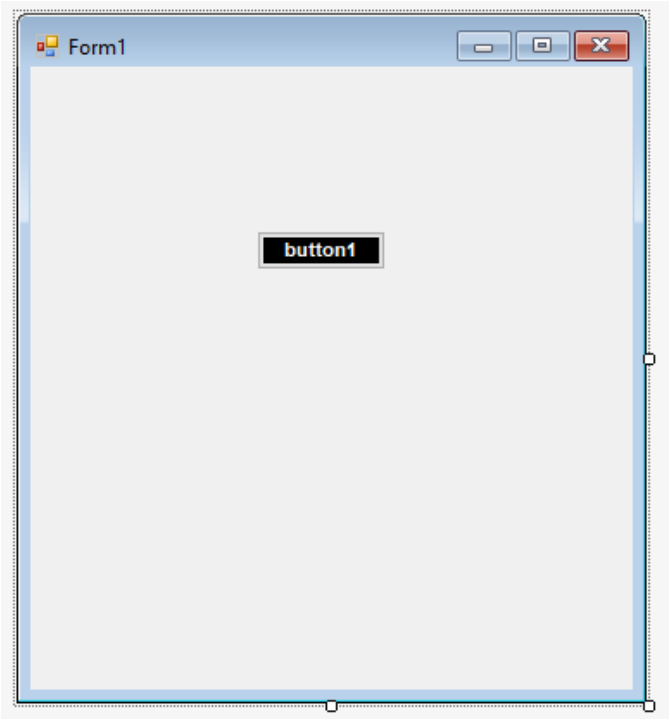
Prelaskom miša preko elementa (engl. *mouse hover*), svaki od elemenata prikazanih na slici 4.3. pružaju kratak opis za što služe i što omogućuju u obrascu. Element se jednostavno prenese na otvoreni obrazac klikom miša na željeni element te povlačenjem i ispuštanjem elementa na željeno mjesto. Jednom kada je element prenesen odnosno dodan unutar obrasca, korisnik ga može kopirati, ukloniti i dodati mu razne već spomenute postavke ili događaje.

Iako je dosad opisani način najčešći za dodavanje elemenata u obrazac ovo nije ujedno i jedini način. Drugi način za dodavanje elemenata je znatno kompliciraniji i on uključuje ručno upisivanje svih podataka vezanih za svaki pojedinačni element u obrascu. Takav kod se upisuje u Form Designer koji je automatski generiran prilikom otvaranja predloška Windows Forms.

U nastavku prikazan je dio programskog koda koji se nalazi u Form Designeru (Programski kod 4.1.), a rezultat pisanja takvog koda nalazi se na slici 4.4.

```
this.button1.BackColor = System.Drawing.SystemColors.ActiveCaptionText;
this.button1.Font = new System.Drawing.Font("Arial", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(238)));
this.button1.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
this.button1.Location = new System.Drawing.Point(131, 95);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(75, 23);
this.button1.TabIndex = 0;
this.button1.Text = "button1";
this.button1.UseVisualStyleBackColor = false;
```

Programski kod 4.1. Primjer koda za element gumb



Slika 4.4. Rezultat programskog koda 4.1.

Ručnim dodavanjem elementa gumb iz sekcije s alatima (engl. *toolbox*) sva prikazana uređenja gumba možemo napraviti u svega nekoliko sekundi dok upisivanje koda u dizajneru bi zahtijevalo ne samo više vremena, nego i bolje poznavanje svih postavki svakog pojedinog elementa.

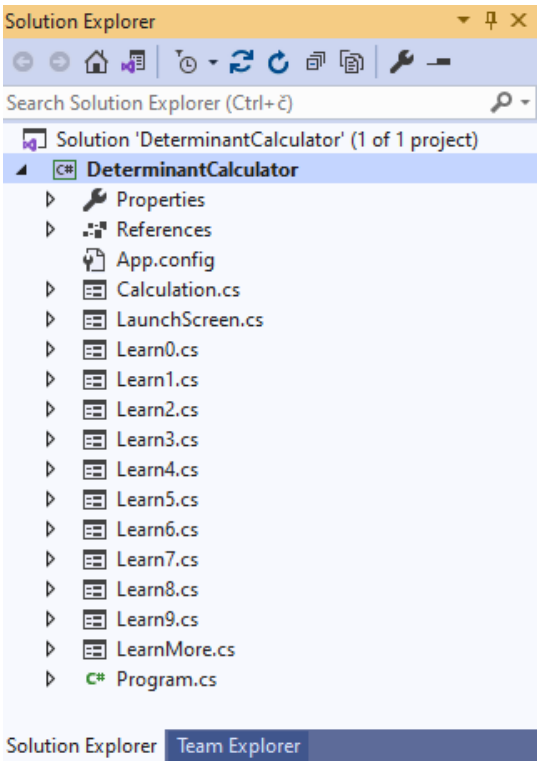
Uspoređujući dva načina za dodavanje elemenata, dolazi se do zaključka da postavljanje kontrola povlačenjem i ispuštanjem elemenata u obrazac je znatno učinkovitije od ručnog pisanja koda.

5. IZRADA APLIKACIJE

Aplikacija se sastoji od dva funkcionalna dijela - informativni dio koji korisniku objašnjava teoriju i primjere zadataka vezanih za rješavanje determinante i glavni dio aplikacije koji omogućuje unos reda i elemenata matrice pomoću kojih se ispisuje determinanta. Navedena dva dijela aplikacije zajedno s početnom stranicom stvoreni su kao obrasci u predlošku Windows Forms. U nastavku je opisan postupak stvaranja obrazaca od kojih se aplikacija sastoji.

Pokretanjem Visual Studija i odabirom Windows Forms predloška, stvoren je projekt pod imenom *DeterminantCalculator*. U projekt su dodani obrasci (engl. *forms*) pomoću kojih će se oblikovati izgled aplikacije. Glavni obrasci su *LaunchScreen* koji prikazuje aplikaciju kada se pokrene, *LearnMore* koja korisniku daje dodatne informacije o determinantama te obrazac *Calculation* koji omogućava korisniku unos reda i elemenata matrice te s tim informacijama računa determinantu i prikazuje ju korisniku.

Sve stavke stvorenog projekta prikazane su na slici 5.1. Dodatno, na slici su prikazani i svi obrasci vezani uz *LearnMore* karticu odnosno *LearnX* obrasci gdje X predstavlja redni broj informativne kartice koja daje primjer ili definiciju vezanu uz determinante.



Slika 5.1. Sadržaj projekta

Pošto *Learn* obrazaca ima deset, svaki od obrazaca nije ručno kreiran, nego je napravljen predložak prvog obrasca odnosno obrasca *Learn0* koji je potom kopiran još devet puta. Predložak je napravljen tako da je prvo potpuno uređen obrazac *Learn0* sa svim elementima i funkcijama. Zatim unutar Visual Studija u izborniku Projekt (engl. *Project*) odabere se Izvoz predloška (engl. *Export Template*). Na prvoj stranici gdje se traži odabir vrste predloška odabire se *Item Template* i odabire se projekt, u ovom slučaju projekt *DeterminantCalculator*. Na sljedećoj stranici za odabir stavke za izvoz odabere se stavku *Learn0* od koje se stvara predložak. Klika se gumb *Dalje* dok se ne dođe do stranice za odabir mogućnosti predloška gdje se unese naziv predloška i po izboru opis, sliku ikone i sliku pregleda, a zatim se odabere gumb *Završi*. Datoteke za predložak dodaju se u .zip datoteku i kopiraju u odabrani direktorij. Visual Studio se ponovno pokrene, a prilikom dodavanja stavki u projekt nalazi se i napravljeni predložak. [17]

5.1. Obrazac *LaunchScreen*

Obrazac *LaunchScreen* prikazuje početni izgled aplikacije i omogućuje otvaranje drugih obrazaca. Programski kod 5.1. prikazuje kod ovog obrasca.

```
namespace DeterminantCalculator
{
    public partial class LaunchScreen : Form
    {
        public LaunchScreen(){...}

        private void LearnButton_MouseEnter(object sender, EventArgs e) {...}
        private void LearnButton_MouseLeave(object sender, EventArgs e) {...}
        private void CalculationButton_MouseEnter(object sender, EventArgs e) {...}
        private void CalculationButton_MouseLeave(object sender, EventArgs e) {...}
        private void ExitButton_MouseEnter(object sender, EventArgs e) {...}

        private void ExitButton_MouseLeave(object sender, EventArgs e) {...}
        private void LearnButton_Click(object sender, EventArgs e) {...}
        private void CalculationButton_Click(object sender, EventArgs e) {...}
        private void ExitButton_Click(object sender, EventArgs e) {...}
    }
}
```

Programski kod 5.1. Kod obrasca *LaunchScreen*

Prije pisanja koda, kao naslov aplikacije dodana je slika prikazana na slici 5.2. odnosno logo FERIT-a uz koji je napisan tekst Kalkulator determinante pomoću alata Tekstni okvir (engl. *Label*).

Slika 5.2. Logo FERIT-a

Potom su dodana tri gumba – *LearnButton* koji vodi na obrazac *LearnMore*, *CalculateButton* koji vodi na obrazac *Calculation* i *ExitButton* za izlaz iz aplikacije. Za svaki gumb postavljena je plava boja za pozadinsku boju te bijela boja za tekst. Dodatno, svakom gumbu dodan je događaj (engl. *event*) za prelazak miša preko gumba (engl. *MouseEnter*) koji postavlja pozadinsku boju gumba u crvenu i odlazak miša (engl. *MouseLeave*) koji vraća pozadinsku boju gumba nazad u plavu. Programski kod za navedene funkcije prikazan je u programskom kodu 5.2.

```
private void LearnButton_MouseEnter(object sender, EventArgs e)
{
        LearnButton.BackColor = Color.FromArgb(225, 59, 52);
}

private void LearnButton_MouseLeave(object sender, EventArgs e)
{
        LearnButton.BackColor = Color.FromArgb(27, 83, 123);
}
```

Programski kod 5.2. Funkcije *MouseEnter* i *MouseLeave* na gumbu *LearnButton*

Rezultat programskog koda 5.2. prikazan je na slikama 5.3. i 5.4.

Slika 5.3. *LearnButton* u početnom stanju (funkcija *MouseLeave*)

Slika 5.4. *LearnButton* pri prelasku miša preko gumba (funkcija *MouseEnter*)

Posljednje, gumbovima je dodana funkcija *Click* koja se pokreće klikom miša na gumb. Ova funkcija kod gumbova *LearnButton* i *CalculateButton* zatvara trenutni obrazac i otvara novi obrazac, a u slučaju gumba *ExitButton* u potpunosti zatvara aplikaciju. Programski kod za navedene funkcije prikazan je u programskom kodu 5.3.

```
private void LearnButton_Click(object sender, EventArgs e)
{
    this.Hide();
    LearnMore learnForm = new LearnMore();
    learnForm.Show();
}

private void CalculateButton_Click(object sender, EventArgs e)
{
    this.Hide();
    Calculation calculateForm = new Calculation();
    calculateForm.Show();
}

private void ExitButton_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Programski kod 5.3. Funkcije *LearnButton_Click*, *CalculateButton_Click* i *ExitButton_Click*

5.2. Obrazac *LearnMore*

Obrazac *LearnMore* početni je obrazac za dodatne informacije o determinantama. Programski kod 5.4. prikazuje kod ovog obrasca.

```
namespace DeterminantCalculator
{
    public partial class LearnMore : Form
    {
        public LearnMore() {...}

        private void BackButton_Click(object sender, EventArgs e) {...}
        private void BackButton_MouseEnter(object sender, EventArgs e) {...}
        private void BackButton_MouseLeave(object sender, EventArgs e) {...}

        private void label1_Click(object sender, EventArgs e) {...}
        private void label2_Click(object sender, EventArgs e) {...}
        private void label3_Click(object sender, EventArgs e) {...}
        private void label4_Click(object sender, EventArgs e) {...}
        private void label5_Click(object sender, EventArgs e) {...}
        private void label6_Click(object sender, EventArgs e) {...}
        private void label7_Click(object sender, EventArgs e) {...}
        private void label8_Click(object sender, EventArgs e) {...}
        private void label9_Click(object sender, EventArgs e) {...}
        private void label10_Click(object sender, EventArgs e) {...}
    }
}
```

Programski kod 5.4. Kod obrasca *LearnMore*

Kao što se može zaključiti po programskom kodu 5.4. gledajući imena funkcija, u ovom obrascu nalazi se deset tekstnih okvira (engl. *label*) od kojih svaki vodi na drugi obrazac koji daje korisniku

novu informaciju vezanu za determinante. Npr. tekstni okvir prikazan na slici 5.5. tj. *Determinanta je…* prilikom klika miša otvorit će obrazac s definicijom : Determinanta je skalarna veličina dobivena zbrojem umnožaka elemenata kvadratne matrice i njihovih kofaktora prema propisanom pravilu.

Slika 5.5. Primjer tekstnog okvira

Funkcija za događaj klika mišem koji je postavljen za svaki tekstni okvir u ovom obrascu prikazana je u programskom kodu 5.5.

```
private void label1_Click(object sender, EventArgs e)
{
       this.Hide();
       Learn0 learnForm = new Learn0();
       learnForm.Show();
}
```

Programski kod 5.5. Funkcija za klik na tekstni okvir

Osim tekstnih okvira, u ovom obrascu nalazi se gumb *BackButton* koji ima svijetlo plavu pozadinu, a prilikom prelaska miša preko gumba mijenja pozadinu u svjetliju crvenu. Ovaj gumb nalazi se u svim obrascima osim početnog i u svim obrascima ima isti izgled te nalazi se na istom mjestu. Konkretno u ovom obrascu i u obrascu *Calculation* ovaj gumb vodi na početnu stranicu odnosno na obrazac *LaunchScreen*. Sve funkcije gumba *BackButton* prikazane su u programskom kodu 5.6. tj. funkcije *BackButton_Click*, *BackButton_MouseEnter* i *BackButton_MouseLeave.*

```
private void BackButton_Click(object sender, EventArgs e)
{
    this.Hide();
    LaunchScreen launchForm = new LaunchScreen();
    launchForm.Show();
}

private void BackButton_MouseEnter(object sender, EventArgs e)
{
    BackButton.BackColor = Color.FromArgb(231, 114, 110);
}
private void BackButton_MouseLeave(object sender, EventArgs e)
{
    BackButton.BackColor = Color.FromArgb(153, 180, 209);
}
```

Programski kod 5.6. Funkcije gumba *BackButton*

5.3. Obrasci *Learn*

Learn obrasci daju primjere i definicije vezane uz determinante. U imenu ovih obrazaca je i broj od 0 do 9 pošto se radi o ukupno deset obrazaca. Svaki od *Learn* obrazaca povezan je s obrascem *LearnMore* na način da ako se *LearnMore* ne otvori ne može se doći ni do *LearnX* obrasca. Npr. obrazac *Learn0* otvorit će se klikom na 1. tekstni okvir u obrascu *LearnMore*, klikom na 2. tekstni okvir otvorit će se obrazac *Learn1* itd. sve do zadnjeg tekstnog okvira koji će otvoriti obrazac *Learn9*.

Programski kod 5.7. prikazuje kod obrasca *Learn0*, a pošto su svi *Learn* obrasci jednaki svi imaju iste funkcije.

```
namespace DeterminantCalculator
{
    public partial class Learn0 : Form
    {
        public Learn0()

        private void BackButton_Click(object sender, EventArgs e)
        private void PreviousButton_Click(object sender, EventArgs e)
        private void NextButton_Click(object sender, EventArgs e)

        private void NextButton_MouseEnter(object sender, EventArgs e)
        private void NextButton_MouseLeave(object sender, EventArgs e)
        private void PreviousButton_MouseEnter(object sender, EventArgs e)
        private void PreviousButton_MouseLeave(object sender, EventArgs e)
        private void BackButton_MouseEnter(object sender, EventArgs e)
        private void BackButton_MouseLeave(object sender, EventArgs e)
    }
}
```

Programski kod 5.7. Kod obrasca *Learn0*

Svaki se obrazac sastoji od gumba *BackButton* koji je već spomenut u prethodnom obrascu, grupe alata (engl. *GroupBox*) u kojem se nalazi neka od definicija ili primjera u obliku tekstnih okvira te dva gumba: *PreviousButton* i *NextButton* koji omogućuju listanje *Learn* obrazaca. Za razliku od obrazaca *LearnMore* i *Calculation*, gumb *BackButton* ne vodi na početnu stranicu aplikacije, nego na obrazac *LearnMore* pošto je to prethodna stranica ovog obrasca što je prikazano u programskom kodu 5.8.

```
private void BackButton_Click(object sender, EventArgs e)
{
    this.Hide();
    LearnMore launchForm = new LearnMore();
    launchForm.Show();
}
```

Programski kod 5.8. Funkcija gumba *BackButton*

Gumbovi *PreviousButton* i *NextButton* sadrže funkcije *MouseEnter* i *MouseLeave* kao što je prikazano na slikama 5.3. i 5.4. odnosno dodan je kod kao u programskom kodu 5.2. Također, sadrže funkcije koje će korisnika odvesti na drugi obrazac odnosno funkcije za događaj klika miša (engl. *Click*). Ove su funkcije prikazane u programskom kodu 5.9.

```
private void PreviousButton_Click(object sender, EventArgs e)
{
    this.Hide();
    Learn9 LearnForm = new Learn9();
    LearnForm.Show();
}

private void NextButton_Click(object sender, EventArgs e)
{
    this.Hide();
    Learn1 LearnForm = new Learn1();
    LearnForm.Show();
}
```

Programski kod 5.9. Funkcije gumbova *PreviousButton* i *NextButton*

5.4. Obrazac *Calculation*

Obrazac *Calculation* omogućava korisniku unos reda matrice. Pomoću te informacije generira se matrica čiji je svaki element zapravo alat tekstni okvir (engl. *Textbox*). Korisnik tada može unijeti vlastitu matricu na osnovi koje se računa determinanta.

Ovaj obrazac sadrži četiri gumba: *InputButton*, *CalculateButton*, *NewMatrixButton* i *BackButton*, tekstni okvir za unos reda matrice *InputTextBox* te dodatan tekst za lakše razumijevanje aplikacije poput „Unesi red matrice:“ i „Determinanta upisane matrice je:“

Programski kod 5.10. prikazuje kod obrasca *Calculation*.

```
{
    public partial class Calculation : Form
    {
        public Calculation()
        {
            InitializeComponent();
        }
        private int order;
        private List<TextBox> matrixElements = new List<TextBox>();

        private void InputButton_MouseEnter(object sender, EventArgs e)
        private void InputButton_MouseLeave(object sender, EventArgs e)
        private void BackButton_MouseEnter(object sender, EventArgs e)
        private void BackButton_MouseLeave(object sender, EventArgs e)
        private void CalculateButton_MouseEnter(object sender, EventArgs e)
        private void CalculateButton_MouseLeave(object sender, EventArgs e)
        private void NewMatrixButton_MouseEnter(object sender, EventArgs e)
        private void NewMatrixButton_MouseLeave(object sender, EventArgs e)

        private double ResizeMatrix(double[,] array, int order)
        private double CalculateDeterminant(double[,] array, int order)
        private double[,] ListToArray(List<TextBox> elements, int order)
        private void CreateMatrix(int order)

        private void InputButton_Click(object sender, EventArgs e)
        private void BackButton_Click(object sender, EventArgs e)
        private void CalculateButton_Click(object sender, EventArgs e)
        private void NewMatrixButton_Click(object sender, EventArgs e)
    }
}
```

Programski kod 5.10. Kod obrasca *Calculation*

Svaki gumb sadrži funkcije *MouseEnter* i *MouseLeave* te funkcije koje se pokreću na događaj klika gumba. Klikom na *BackButton* zatvorit će se trenutna kartica i otvorit će se obrazac *LaunchScreen* kao što je prikazano u programskom kodu 5.6. kod obrasca *LearnMore*. Slična je situacija i s gumbom *NewMatrixButton* koji će ponovno otvoriti prazni obrazac *Calculation* što je prikazano u programskom kodu 5.11.

```
private void NewMatrixButton_Click(object sender, EventArgs e)
{
    this.Hide();
    Calculation calculation = new Calculation();
    calculation.Show();
}
```

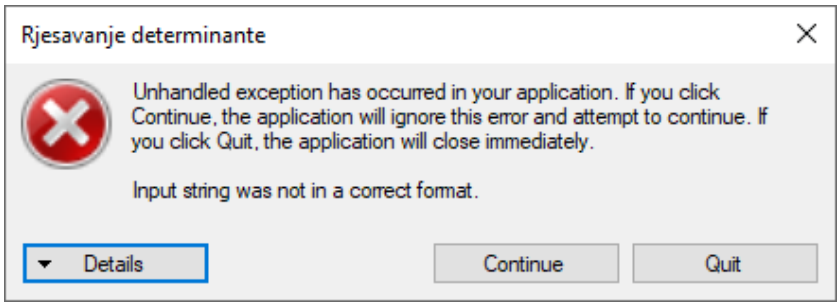
Programski kod 5.11. Funkcija gumba *NewMatrixButton*

Klikom na *InputButton* sprema se uneseni podatak za veličinu reda matrice i ovisno o tom podatku različiti su ishodi aplikacije. Programski kod za ovu funkciju prikazan je u programskom kodu 5.12.

```
private int order;
private void InputButton_Click(object sender, EventArgs e)
{
    order = Int32.Parse(InputTextBox.Text);
    if (order < 1)
    {
       MessageBox.Show("Unesi red matrice veći od 0!", "Greška",
       MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (order > 10)
    {
       MessageBox.Show("Red matrice mora biti manji od 10!", "Greška",
       MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
       CreateMatrix(order);
    }
}
```

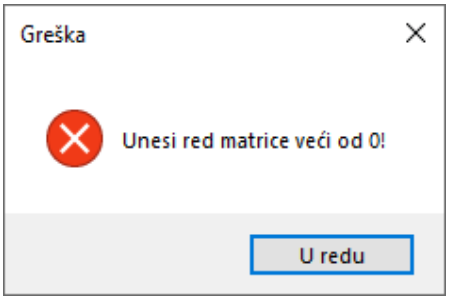
Programski kod 5.12. Funkcije gumba *BackButton*

Kao što prethodni kod prikazuje, postoje tri različita ishoda od kojih svaki ovisi o unesenom redu matrice. Iako kod postavlja tri uvjeta, postoje još dva ishoda koji nisu direktno postavljeni u programskom kodu 5.12. odnosno ako red uopće nije unesen ili ako se unese slovo ili neki drugi znak koji nije broj tipa *int*, prikazat će se pogreška prikazana na slici 5.6.

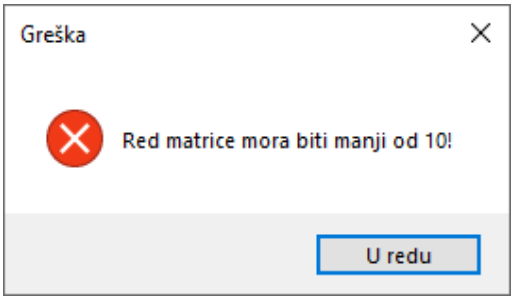


Slika 5.6. Greška pri upisivanju reda matrice

Pošto veličina reda matrice ne može biti manja od 1, postavljen je uvjet i za slučaj upisa takvih brojeva što će izbaciti pogrešku prikazanu na slici 5.7. Slično prethodnom slučaju, postavljen je uvjet pri upisu veličine reda većeg od 10 čija je greška prikazana na slici 5.8.



Slika 5.7. Greška pri upisivanju manjeg reda matrice od dopuštenog



Slika 5.8. Greška pri upisivanju većeg reda matrice od dopuštenog

Posljednje, ukoliko je unesen broj između 1 i 10, pozvat će se funkcija *CreateMatrix* koja će generirati matricu *n*-tog reda gdje je *n* uneseni red matrice. Svaki od elemenata matrice je zapravo tekstni okvir kojemu su unaprijed postavljene sve postavke vezane za izgled i lokaciju. Bitno je napomenuti da se svaki novi element sprema u listu koja će kasnije služiti za računanje determinante. Ova funkcija prikazana je u programskom kodu 5.13.

```
private List<TextBox> matrixElements = new List<TextBox>();
private void CreateMatrix(int order)
{
    for (int row = 0; row < order; row++)
    {

        int counter = 10 - order;
        int x = 180 + counter * 19;
        int y = 129 + row * 38 + counter * 19;
        for (int column = 0; column < order; column++)
        {
            TextBox newElement = new TextBox();
              newElement.Font = new System.Drawing.Font("Arial", 11F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(238)));
              newElement.Location = new System.Drawing.Point(x + column * 38, y);
              newElement.Name = "TextBox1";
              newElement.Size = new System.Drawing.Size(35, 35);
              newElement.TabIndex = 0;
              newElement.Multiline = true;
              newElement.BackColor = SystemColors.ActiveCaption;
              newElement.ForeColor = Color.White;
              newElement.TextAlign = HorizontalAlignment.Center;
              newElement.Text = "0";
              this.Controls.Add(newElement);
              matrixElements.Add(newElement);
       }
    }
}
```

Programski kod 5.13. Funkcija CreateMatrix

Funkcija za događaj klika miša na *CalculateButton* prikazana je u programskom kodu 5.14.

```
private void CalculateButton_Click(object sender, EventArgs e)
{
       double[,] array = ListToArray(matrixElements, order);
       double determinant = CalculateDeterminant(array, order);
       SolutionLabel.Text = $"det(A) = {determinant}";
}
```

Programski kod 5.14. Funkcija gumba *CalculateButton*

Kao što prethodni kod prikazuje, lista u koju su spremljene sve vrijednosti elemenata matrice pretvorena je u polje funkcijom *ListToArray*. Prvo, svi elementi liste se kopiraju u pomoćni niz koji je tipa *TextBox* pošto je tog tipa i sama lista. Zatim u jednodimenzionalno polje tipa *double* kopiramo sve vrijednosti iz pomoćnog niza, ali tako da se sav tekst pretvori u brojeve tipa *double*. Posljednje, iz jednodimenzionalnog polja sve vrijednosti se kopiraju u dvodimenzionalno polje koje će se predati kao argument funkciji *CalculateDeterminant*.

Funkcija *ListToArray* prikazana je u programskom kodu 5.15.

```
private double[,] ListToArray(List<TextBox> elements, int order)
{
    TextBox[] additionalArray = new TextBox[order * order];
    elements.CopyTo(additionalArray);

    double[] array1D = new double[order * order];
    for (int x = 0; x < order * order; x++)
    {
       array1D[x] = Double.Parse(additionalArray[x].Text,
NumberStyles.AllowDecimalPoint | NumberStyles.AllowLeadingSign,
CultureInfo.InvariantCulture);
    }

    double[,] array2D = new double[order, order];
    int y = 0;
    for (int row = 0; row < order; row++)
    {
       for (int column = 0; column < order; column++)
       {
              array2D[row, column] = array1D[y];
              y++;
       }
    }
    return array2D;
}
```

Programski kod 5.15. Funkcija *ListToArray*

5.4.1. Funkcije za računanje determinante

Rješavanje determinante podijeljeno je u dvije funkcije: *ResizeMatrix* i *CalculateDeterminant*.

Funkcija *CalculateDeterminant* računa determinantu tako što provjerava upisani red matrice. Ako je red jednak nula vratiti će vrijednost 0. Ako je red jednak 1 vratiti će vrijednost upisanog elementa matrice pošto za matricu $A = [a]$ vrijedi da je njena determinanta jednaka elementu a prema izrazu (3-6). Zatim, ako je red jednak 2 vratiti će vrijednost determinante po definiciji:

Za matricu drugog reda vrijedi da je njena determinanta jednaka razlici umnoška njenih dijagonala prema izrazu (3-7).

Posljednja provjera je za matrice reda većeg od 2 čime se poziva funkcija *ResizeMatrix* kako bi ju smanjili sve do matrice 2. reda koja se računa po prethodno prikazanom izrazu. Ovaj kod prikazan je u programskom kodu 5.16.

```
private double CalculateDeterminant(double[,] array, int order)
{
    if (order == 0)
       return 0;
    else if (order == 1)
       return array [0, 0];
    else if (order == 2)
       return (array [0, 0] * array [1, 1]) - (array [0, 1] * array [1, 0]);
    else
       return ResizeMatrix(array, order);
}
```

Programski kod 5.16. Funkcija *CalculateDeterminant*

Funkcija *ResizeMatrix* služi za stvaranje podmatrica odnosno zadanu se matricu smanjuje sve dok se ne dođe do matrice 2. reda gdje će determinanta biti riješena prema izrazu (3-7).

Funkcija *ResizeMatrix* prikazana je u programskom kodu 5.17.

```
private double ResizeMatrix(double[,] array, int order)
        {
            double determinant, sign;
            double[] c = new double[11];
            double[,] submatrix = new double[11, 11];

            for (int i = 0; i <= order - 1; i++)
            {
                int row = 0, column = 0;
                for (int j = 0; j <= order - 1; j++)
                {
                    for (int k = 0; k <= order - 1; k++)
                    {
                        if (j != 0 && k != i)
                        {
                            submatrix [row, column] = array[j, k];
                            column++;
                            if (column > order - 2)
                            {
                                row++;
                                column = 0;
                            }
                        }
                    }
                }
                sign = 1;
                for (int t = 0; t <= (1 + i); t++)
                {
                    sign = (-1) * sign;
                }
                c[i] = sign * CalculateDeterminant(submatrix, order - 1);
            }

            determinant = 0;
            for (int i = 0; i <= order - 1; i++)
            {
                determinant = determinant + (array[0, i] * c[i]);
            }
            return (determinant);
        }
```

Programski kod 5.17. Funkcija *ResizeMatrix*

U funkciji *ResizeMatrix* korištenjem for petlji postavljaju se elementi podmatrice, određuje se predznak i ponovno se poziva funkcija *CalculateDeterminant*. Sve navedeno najbolje je objasniti kroz primjer.

Primjer 5.1. Rješavanje determinante matrice 3. reda.

Zadana je matrica A:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

1. iteracija kada je $i = 0$ prikazana je u tablici 5.1.

Tablica 5.1. Prva iteracija

i	0	0	0	0	0	0	0	0	0
j	0	0	0	1	1	1	2	2	2
k	0	1	2	0	1	2	0	1	2

Brojevi j i k predstavljaju redak i stupac prvotne matrice i pomoću njih stvaraju se elementi podmatrice. Broj i također predstavlja stupac početne matrice, ali taj podatak je vezan za razvoj matrice po 1. retku. Konkretno u ovom slučaju stvara se podmatrica koja se dobije izbacivanjem prvog retka i prvog stupca početne matrice odnosno:

$$podmatrica = \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix} \rightarrow det = -3$$

Ova podmatrica dobivena je tako što su iz tablice 5.1. samo narančasto označeni stupci prošli uvjet zadan u programskom kodu 5.17. Time se postavljaju vrijednosti elemenata podmatrice na način:

$$podmatrica[0,0] = A[1,1] = 5$$
$$podmatrica[0,1] = A[1,2] = 6$$
$$podmatrica[1,0] = A[2,1] = 8$$
$$podmatrica[1,1] = A[2,2] = 9$$

Unutar iteracije određuje se i predznak koji će u ovom slučaju biti pozitivan.

2. iteracija kada je $i = 1$ prikazana je u tablici 5.2.

Tablica 5.2. Druga iteracija

i	1	1	1	1	1	1	1	1	1
j	0	0	0	1	1	1	2	2	2
k	0	1	2	0	1	2	0	1	2

U ovoj iteraciji predznak će biti negativan te stvara se podmatrica koja se dobije izbacivanjem prvog retka i drugog stupca početne matrice odnosno:

$$podmatrica = \begin{bmatrix} 4 & 6 \\ 7 & 9 \end{bmatrix} \rightarrow det = -6$$

3. iteracija kada je *i* = 2 prikazana je u tablici 5.3.

Tablica 5.3. Treća iteracija

i	2	2	2	2	2	2	2	2	2
j	0	0	0	1	1	1	2	2	2
k	0	1	2	0	1	2	0	1	2

U ovoj iteraciji predznak će biti pozitivan i stvara se podmatrica koja se dobije izbacivanjem prvog retka i drugog stupca početne matrice odnosno:

$$podmatrica = \begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix} \rightarrow det = -3$$

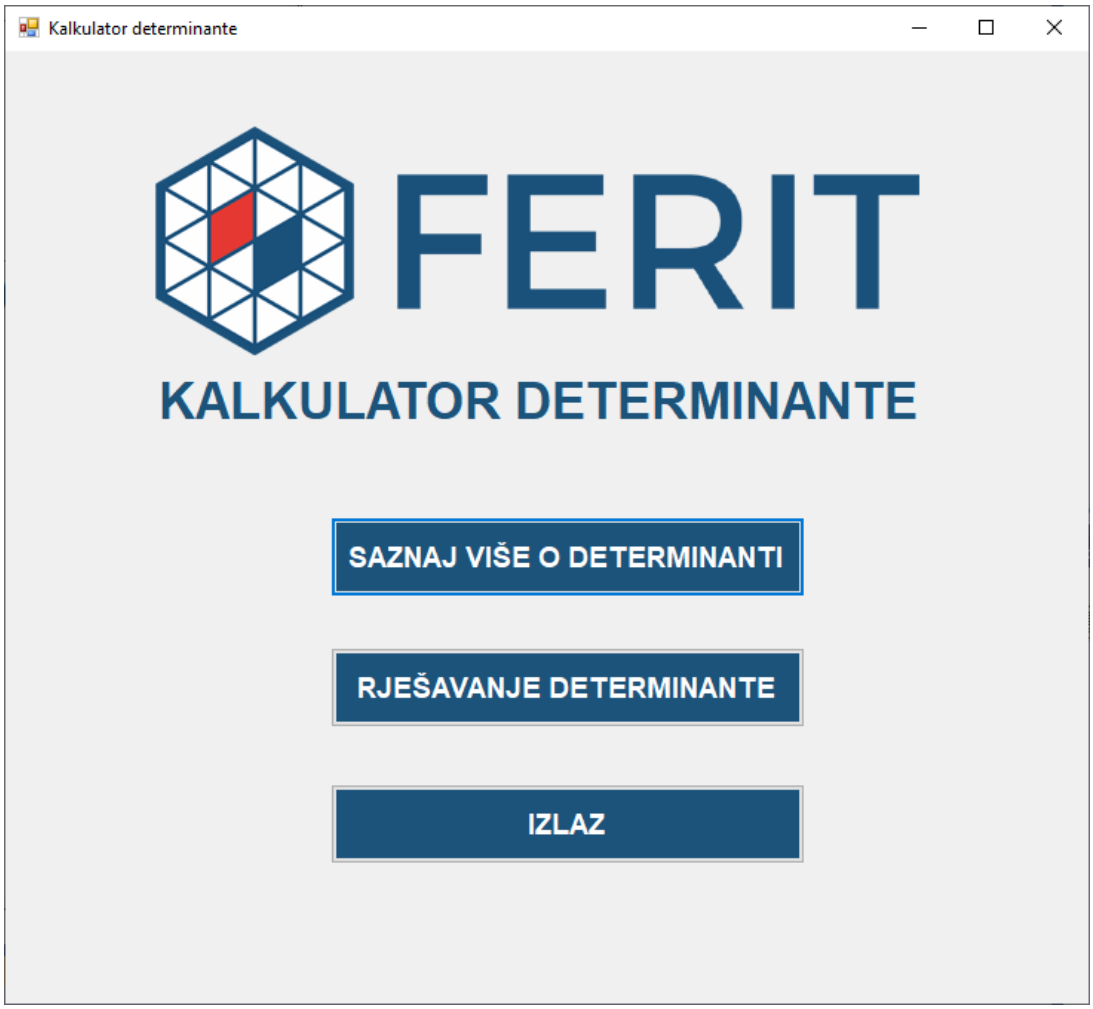
Kada se prođu sve iteracije zbrajaju se sve dobivene determinante koje su dodatno pomnožene s odgovarajućim elementom iz početne matrice kako bi konačno dobili determinantu početne matrice *A* odnosno:

$$\det(A) = 1 \cdot (-3) - 2 \cdot (-6) + 3 \cdot (-3) = 0$$

6. PRIKAZ KORIŠTENJA APLIKACIJE

Spomenuto je da se aplikacija sastoji od ukupno 13 obrazaca od kojih su tri glavna obrasca i 10 sličnih informativnih obrazaca. Pri pokretanju aplikacije otvoren je obrazac *LaunchScreen* preko kojeg se dolazi do svih ostalih obrazaca.

Konačni izgled obrasca *LaunchScreen* prikazan je na slici 6.1.

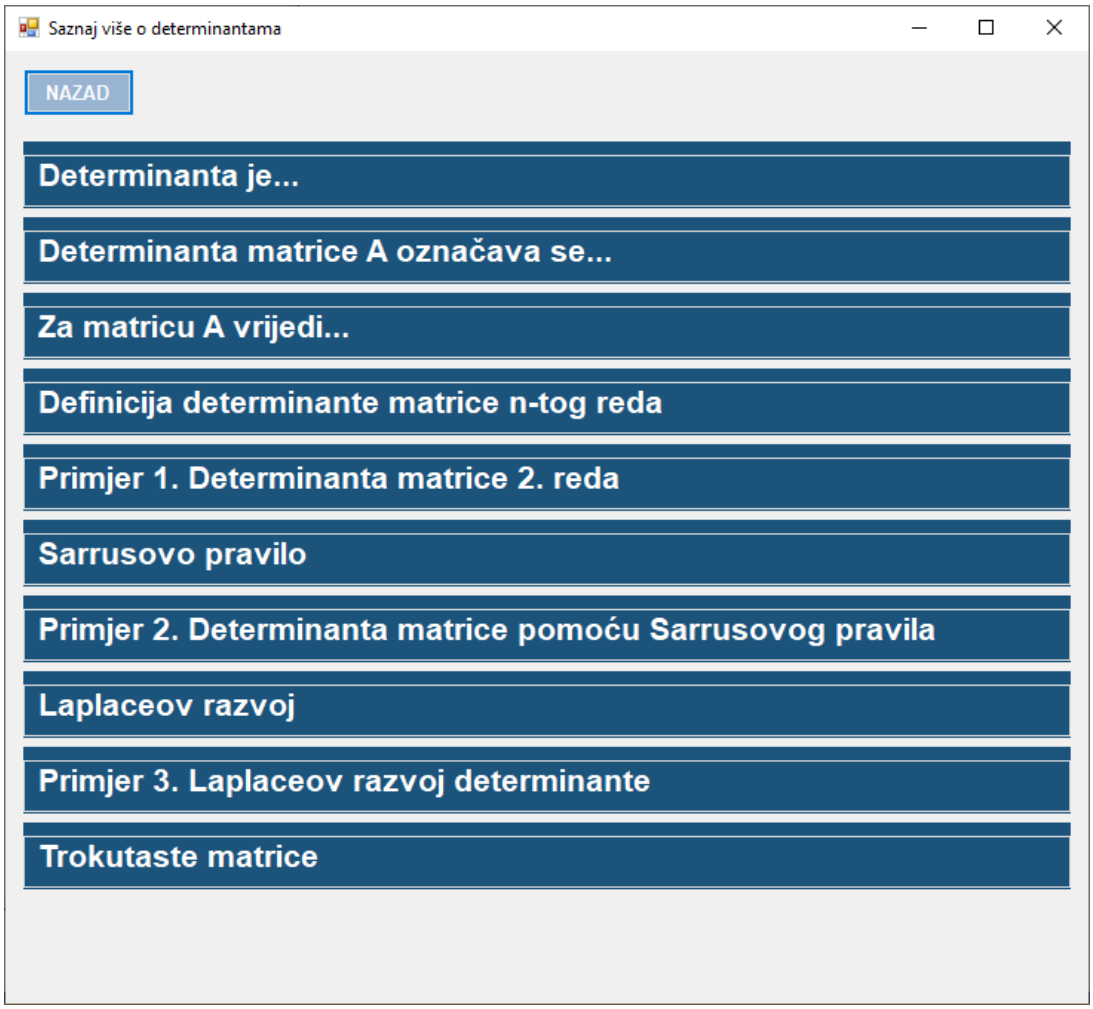


Slika 6.1. Početna stranica aplikacije

Klikom na gumb *Saznaj više o determinanti* prelazi se na obrazac *LearnMore* koji pruža samo početni dio informacija koje će dodatno biti objašnjene u *Learn* obrascima.

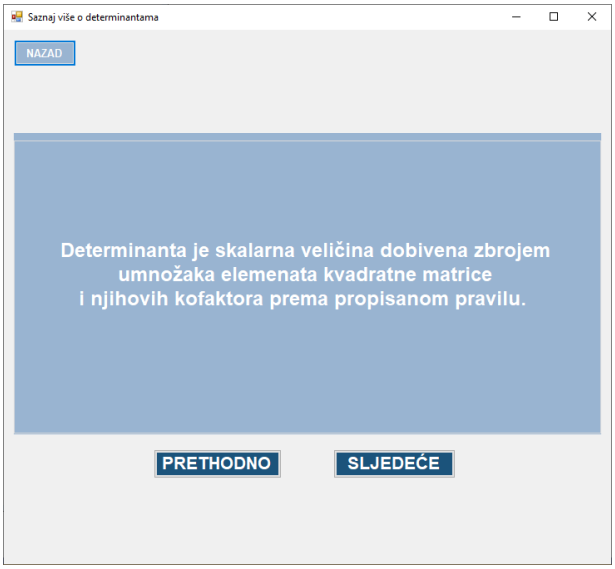
Konačni izgled obrasca *LearnMore* prikazan je na slici 6.2.

Klikom na gumb *Rješavanje determinante* otvara se obrazac *Calculation* gdje korisnik upisuje matricu i dobiva njenu determinantu.

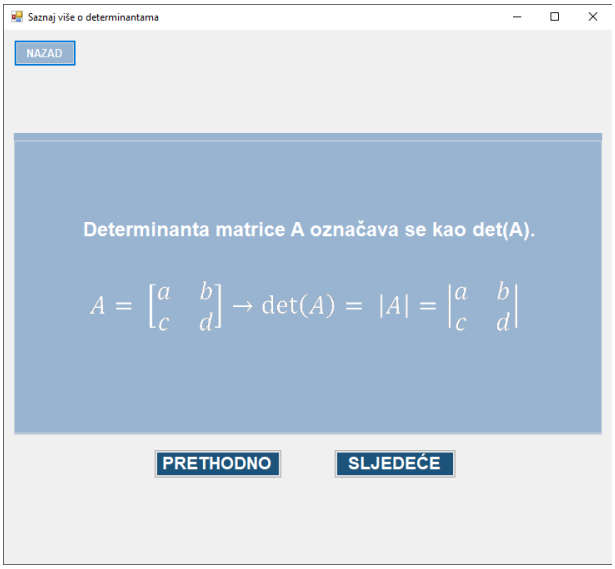


Slika 6.2. Izgled obrasca *LearnMore*

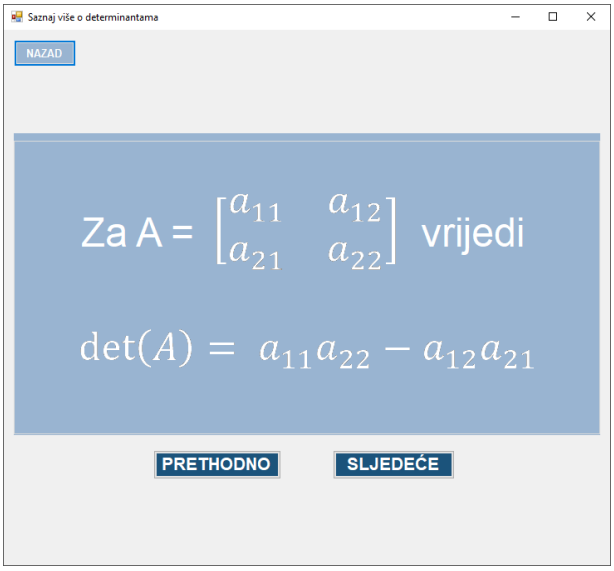
Learn obrasci korisniku daju dodatne informacije o determinantama poput definicija, izraz i primjera rješavanja. U nastavku su prikazane slike 6.3. – 6.12. koje prikazuju *Learn* obrasce.



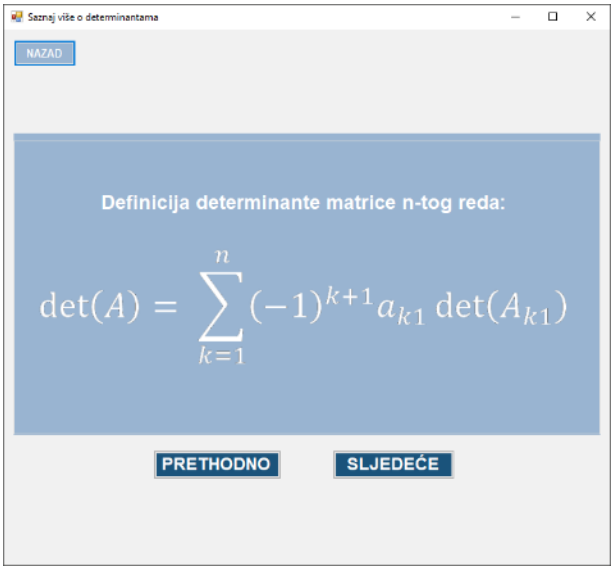
Slika 6.3. Izgled obrasca *Learn0*



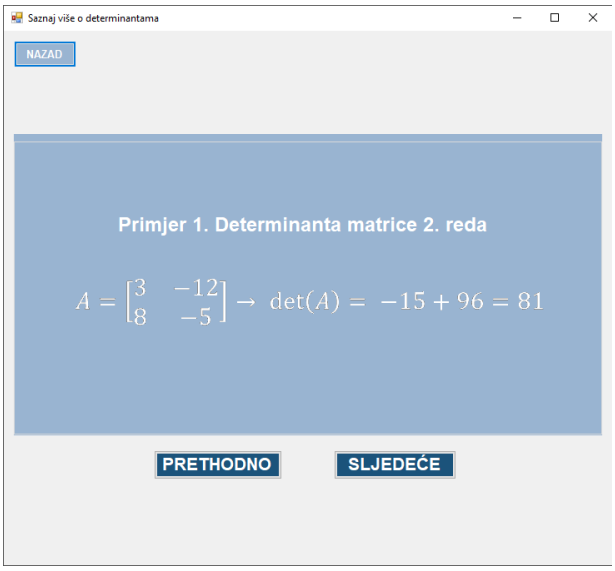
Slika 6.4. Izgled obrasca *Learn1*



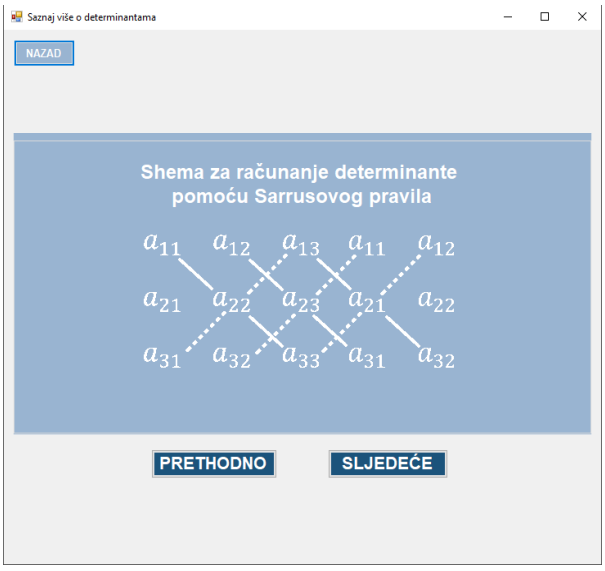
Slika 6.5. Izgled obrasca *Learn2*



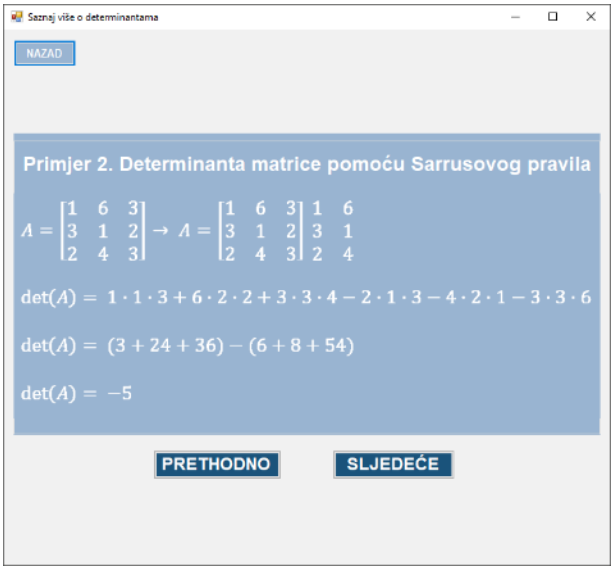
Slika 6.6. Izgled obrasca *Learn3*



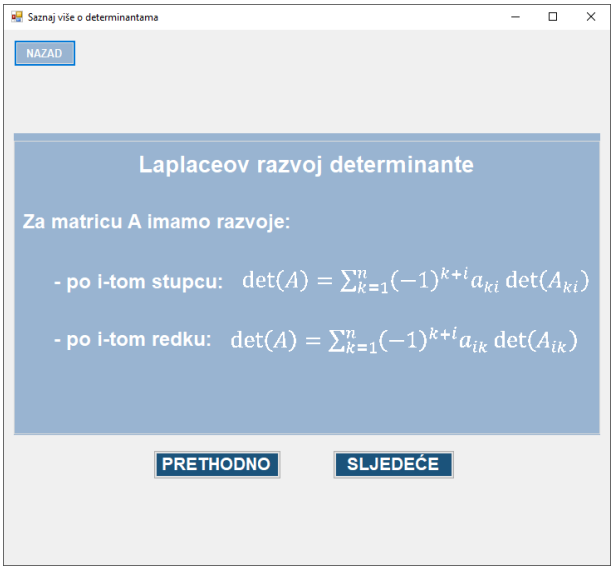
Slika 6.7. Izgled obrasca *Learn4*



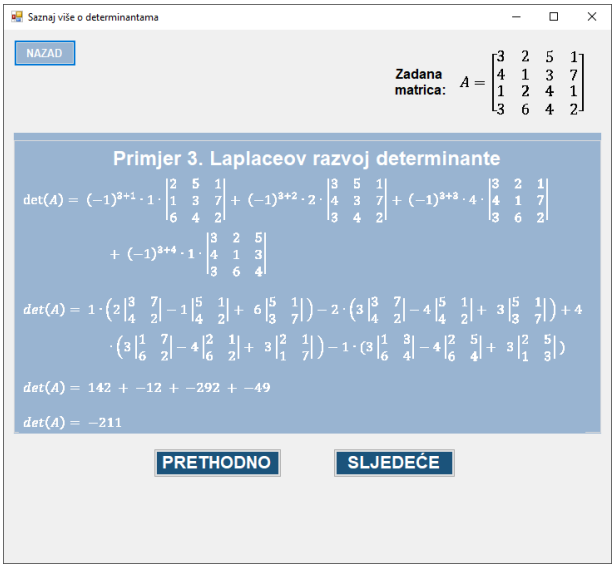
Slika 6.8. Izgled obrasca *Learn5*



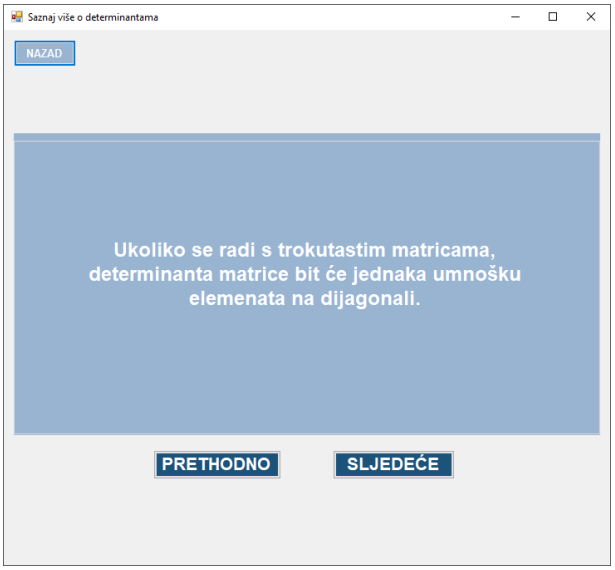
Slika 6.9. Izgled obrasca *Learn6*



Slika 6.10. Izgled obrasca *Learn7*

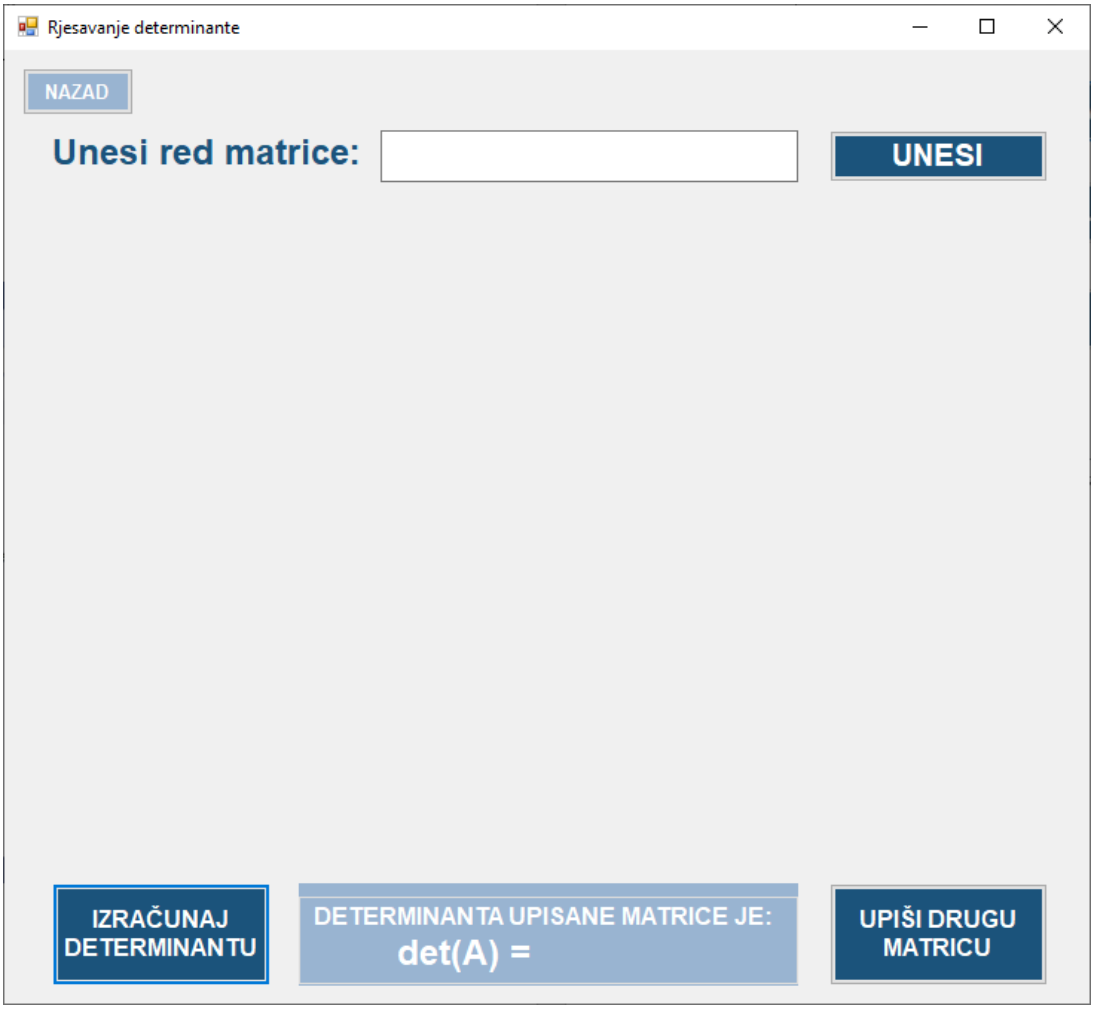


Slika 6.11. Izgled obrasca *Learn8*



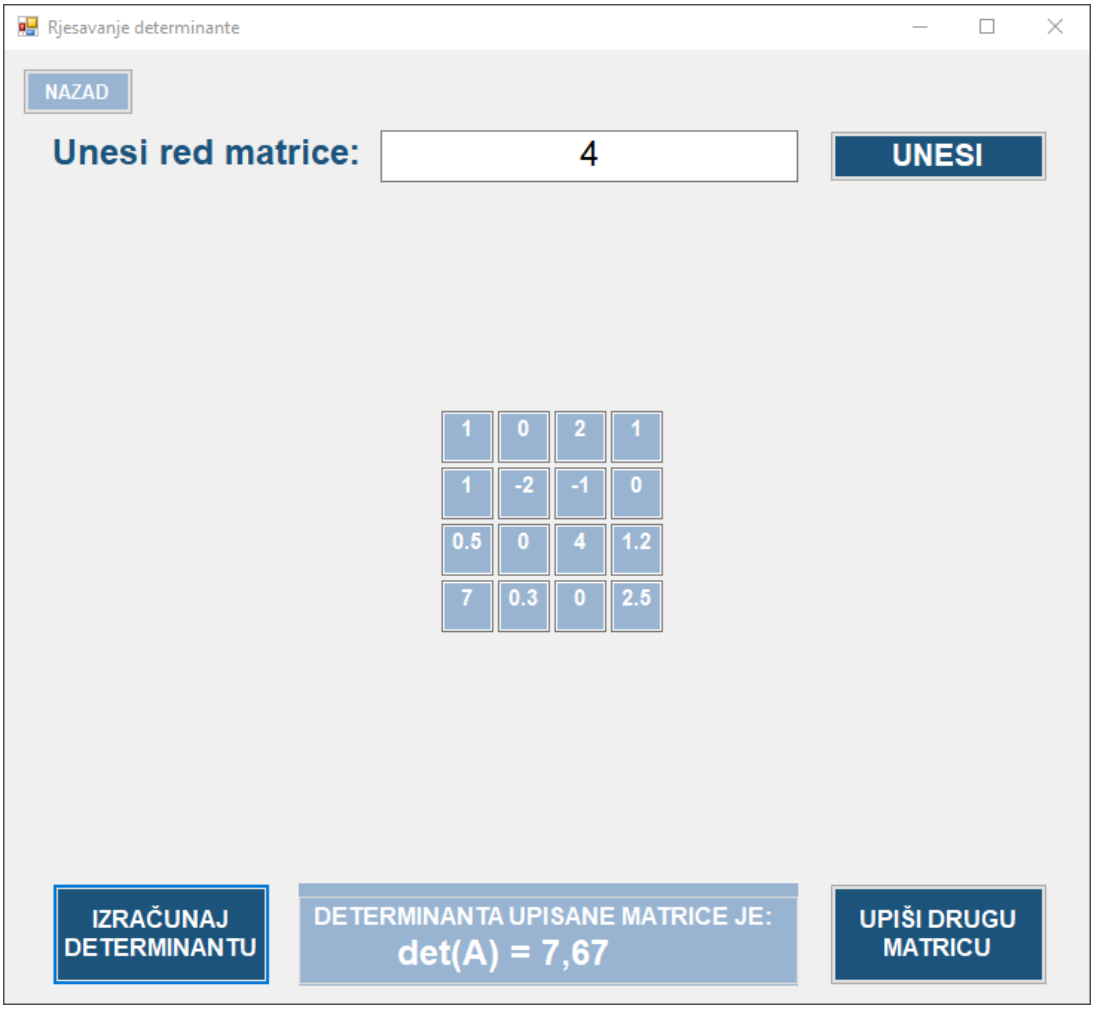
Slika 6.12. Izgled obrasca *Learn9*

Početni izgled obrasca *Calculation* prikazan je na slici 6.13.



Slika 6.13. Izgled obrasca *Calculation*

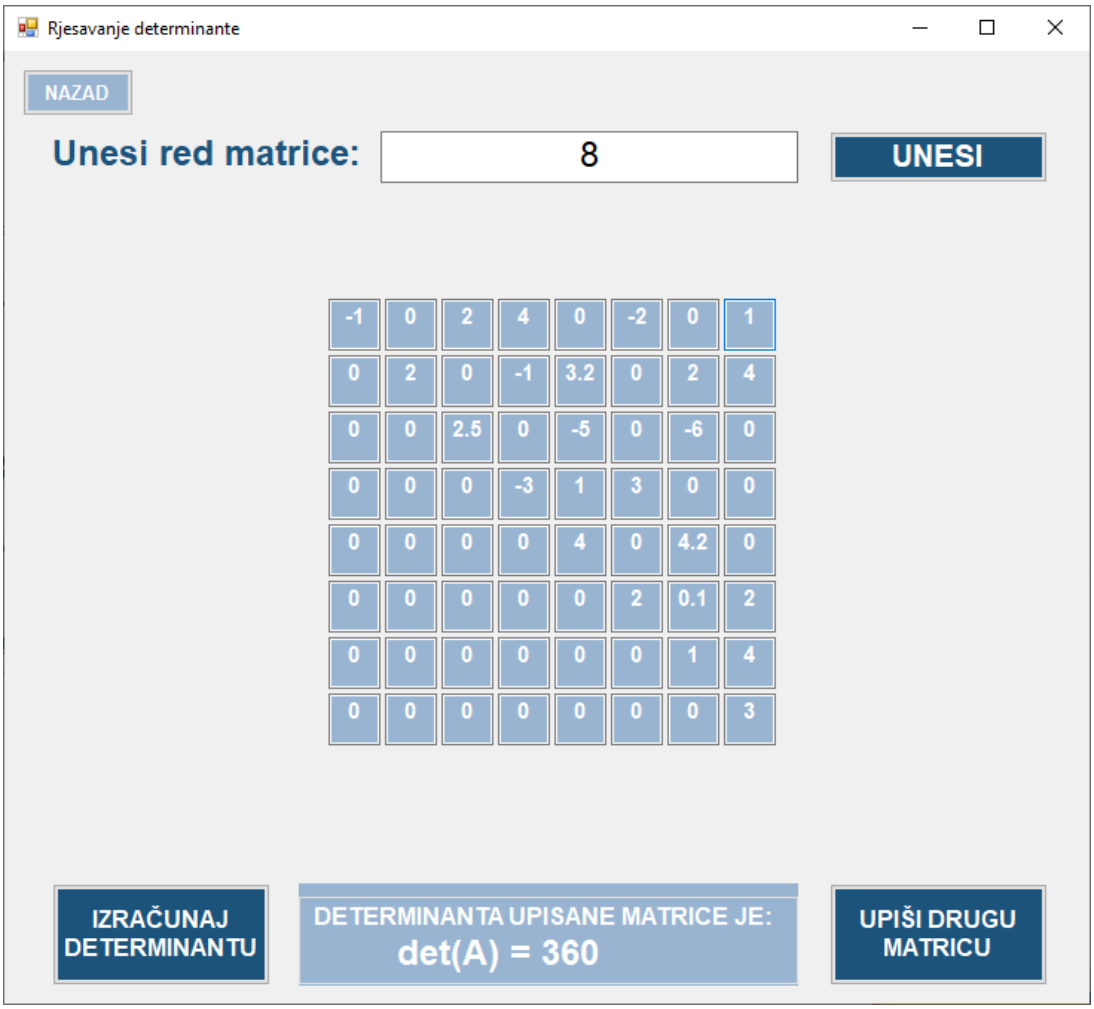
Na slici 6.14. prikazan je 1. primjer riješene determinante i to za matricu 4. reda. Dakle, korisnik je prvo morao upisati red matrice i kliknuti na gumb *Unesi*. Potom upisuje sve elemente matrice i klika na gumb *Izračunaj determinantu* koji zatim ispisuje vrijednost determinante zadane matrice.



Slika 6.14. Primjer riješene determinante matrice 4. reda

Klikom na gumb *Upiši drugu matricu* briše se prethodno napisan red matrice i svi elementi posljednje matrice što uključuje i prozore za upisivanje elemenata matrice tj. aplikacija poprimi izgled koji je prikazan na slici 6.13.

Slika 6.15. prikazuje primjer riješene determinante za matricu 8. reda

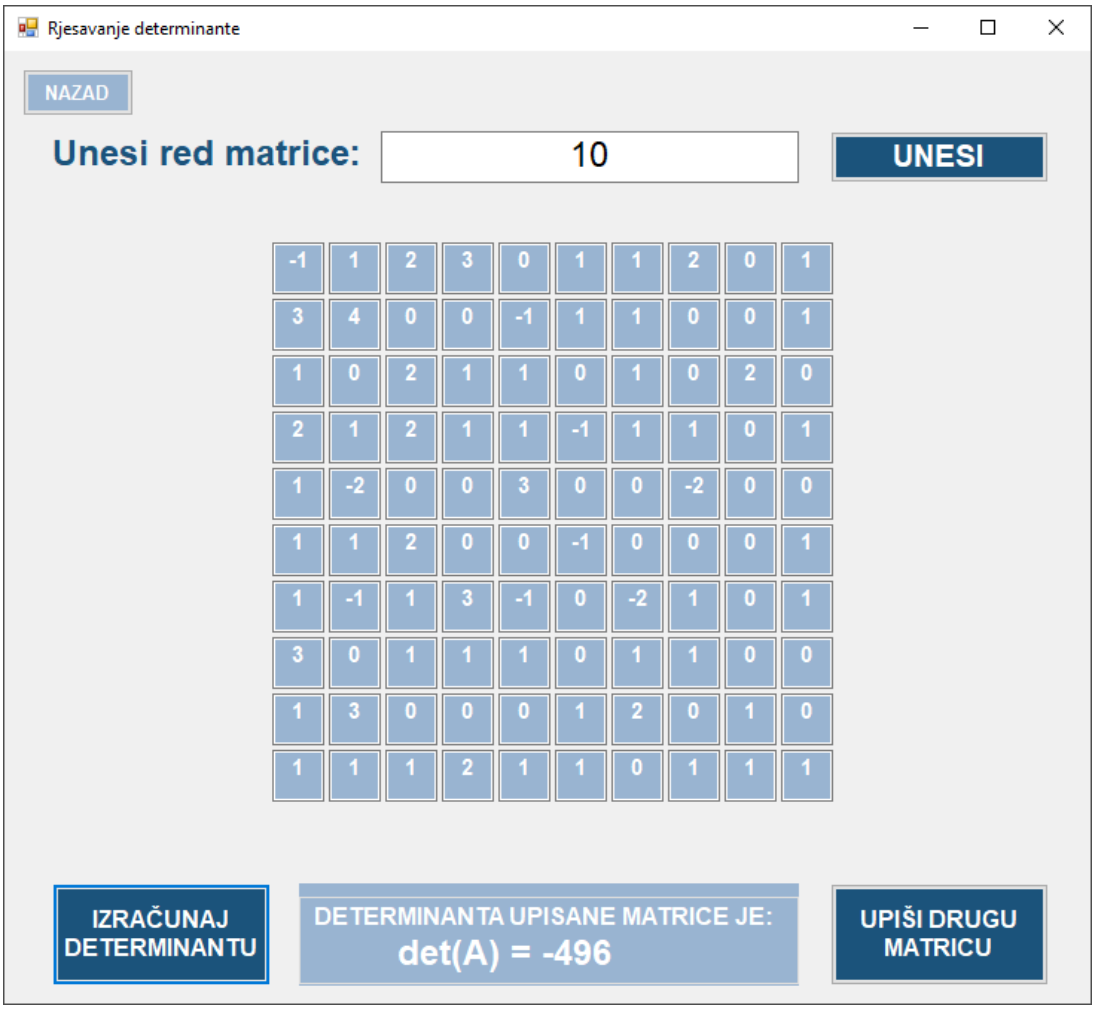


Slika 6.15. Primjer riješene determinante matrice 8. reda

Gledajući upisanu matricu lako se primijeti da se radi o gornje trokutastoj matrici. Prateći pravilo iz obrasca *Learn9* odnosno svojstvo prikazano na slici 6.12. jednostavno se provjeri točnost prikazane determinante matrice 8. reda. Dakle, provjerava se umnožak elemenata na dijagonali:

$$\det(A) = -1 \cdot 2 \cdot 2.5 \cdot (-3) \cdot 4 \cdot 2 \cdot 1 \cdot 3 = 360$$

Slika 6.16. prikazuje primjer riješene determinante matrice 10. reda što je ujedno i najveći red koji se može upisati.



Slika 6.16. Primjer riješene determinante matrice 10. reda

7. ZAKLJUČAK

Na početku rada opisana su postojeća rješenja s istom namjenom. Obrađena je i teorijska osnova rada kroz definicije, svojstva i primjere rješavanja determinanti. Također, spomenute su korištene tehnologije i ukratko je svaka od njih opisana. Osim toga, objašnjen je postupak izrade aplikacije na način da je svaki obrazac pojedinačno razrađen što je uključivalo prikaz svih funkcija obrasca, objašnjenje korištenih funkcija, opis izgleda obrasca i općenite funkcionalnosti. Zaključno, prikazan je izgled cjelokupne aplikacije gdje je prikazan svaki obrazac i dodatno su postavljeni primjeri korištenja aplikacije odnosno primjeri riješenih determinanti matrica različitih redova.

Aplikacija se može doraditi tako da se korisniku omogući upis većeg reda matrice pošto je trenutno ograničena na matrice do desetog reda. Osim toga, bilo bi korisno dodati više primjera i općenito informacija vezane za determinante, ali isto tako i informacije vezane za matrice jer na taj način korisnik bi bolje shvatio teoriju determinanti. Još neke od dodatnih funkcionalnosti mogle bi biti automatsko upisivanje nasumične matrice ili automatsko popunjavanje praznih mjesta u matrici nekim određenim brojem. Svi navedeni nedostaci odnosno proširenja rada lako se mogu nadodati u aplikaciju pogotovo proširenje teorijskih obrazaca. Jedini zahtjevniji dodatak bio bi upis većeg reda matrice pošto zahtjeva promjenu većeg dijela koda za generiranje matrice i sučelje aplikacije trebalo bi se također povećati što bi moglo dovesti do promjene izgleda drugih obrazaca.

LITERATURA

[1] Symbolab, 2023, dostupno na: https://www.symbolab.com/

[2] Symbolab: Math Problem Solver, Google Play, 2023, dostupno na: https://play.google.com/store/apps/details?id=com.devsense.symbolab&hl=en&gl=US [21.3.2023.]

[3] Matrix Determinant Calculator, Google Play, 2023, dostupno na: https://play.google.com/store/apps/details?id=an.DeterminantX&hl=en&gl=US [21.3.2023.]

[4]I. Slapničar, Matematika 1, 2018, dostupno na: http://www.mathematics.digital/matematika1/predavanja/node41.html [23.1.2023.]

[5] The determinant of a matrix, 2022, dostupno na: https://mathinsight.org/determinant_matrix [17.2.2022.]

[6] Rule of Sarrus, Wikipedia, 2023, dostupno na: https://en.wikipedia.org/wiki/Rule_of_Sarrus [3.2.2023.]

[7] K. Horvatić, Linearna algebra, Golden marketing - Tehnička knjiga, Zagreb, 2004.

[8]M, Taboga, "The Laplace expansion, minors, cofactors and adjoints", Lectures on matrix algebra, 2021, dostupno na: https://www.statlect.com/matrix-algebra/Laplace-expansion-minors-cofactors-adjoints [23.1.2023.]

[9] Mathematics | LU Decomposition of a System of Linear Equations, geeksforgeeks, 2021, dostupno na: https://www.geeksforgeeks.org/l-u-decomposition-system-linear-equations/ [3.2.2023.]

[10] I. Slapničar, Matematika 1, 2018, dostupno na: http://www.mathematics.digital/matematika1/predavanja/node32.html [21.3.2023.]

[11] K. Kuttler, Applications of the Determinant, 2022, dostupno na: https://math.libretexts.org/Bookshelves/Linear_Algebra/A_First_Course_in_Linear_Algebra_(Kuttler)/03%3A_Determinants/3.04%3A_Applications_of_the_Determinant#title [2.2.2023.]

[12] C. D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM, 2000.

[13] P. Waghmare, Applications of Matrices and Determinants with Solved Examples, 2023, dostupno na: https://testbook.com/learn/maths-applications-of-matrices-and-determinants/ [17.2.2023.]

[14] What is C# Programming? A Beginner's Guide, 2022, dostupno na: https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c- [23.1.2023.]

[15] Visual Studio, Microsoft, 2023, dostupno na https://visualstudio.microsoft.com [23.1.2023.]

[16] Windows Forms, Techopedia, 2023, dostupno na: https://www.techopedia.com/definition/24300/windows-forms-net [17.2.2022.]

[17] How to: Create item templates, Microsoft, 2022, dostupno na: https://learn.microsoft.com/en-us/visualstudio/ide/how-to-create-item-templates?view=vs-2019 [7.2.2022.]

SAŽETAK

U ovom radu definira se determinanta, objašnjena su njena svojstva, načini rješavanja i primjena. Potom je opisano programsko okruženje Visual Studio unutar kojeg je korišten predložak Windows Forms za stvaranje aplikacije na temelju obrazaca. Detaljno je objašnjen postupak izrade aplikacije te na kraju rada prikazan je konačan izgled i primjeri korištenja aplikacije. Aplikacija od korisnika traži unos reda i elemenata matrice pomoću kojih računa determinantu, a dodatno pruža korisniku i nekoliko teorijskih zanimljivosti o determinanti.

Ključne riječi: C#, determinanta, matrica, Visual Studio, Windows Forms

ABSTRACT

This paper defines the determinant, it explains its properties, solutions and application. Then the Visual Studio programming environment was described, within which the Windows Forms template was used to create an application based on forms. The process of creating the application is explained in detail, and at the end of the paper, the final layout and examples of using the application are shown. The application asks the user to enter the order and elements of the matrix with which it calculates the determinant, and additionally provides the user with several theoretical facts about the determinant.

Keywords: C#, determinant, matrix, Visual Studio, Windows Forms